

Peer-to-Peer Computing for Mobile Networks

*Information Discovery and
Dissemination*

Peer-to-Peer Computing for Mobile Networks

Information Discovery and Dissemination

by

Maria Papadopouli

Department of Computer Science

University of Crete

&

Institute of Computer Science

Foundation for Research and Technology-Hellas (FORTH)

Crete, Greece

Henning Schulzrinne

Department of Computer Science

Columbia University

NY, USA



Springer

Authors:

Maria Papadopouli
Department of Computer Science
University of Crete
Knossos Ave. P.O. Box 2208
Heraklion, GR-714 09 Crete, Greece
&
Institute of Computer Science
Foundation for Research and Technology-Hellas (FORTH)
N. Plastira 100, Vassilika Vouton,
Heraklion, GR-700 13, Crete, Greece
mgp@ics.forth.gr

Henning Schulzrinne
Department of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027, USA
hgs@cs.columbia.edu

Library of Congress Control Number: 2008934312

ISBN-13: 978-0-387-24427-3

e-ISBN-13: 978-0-387-68504-5

DOI: 10.1007/b110083

© 2009 Springer Science+Business Media, LLC.

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden. The use in this publication of trade names, trademarks, service marks and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper.

springer.com

This work is a metaphorical bridge for me, bringing together research results obtained, while I was in the following academic institutions:

- Computer Science Department of Columbia University, as a Ph.D. student (1996–2002)
- Department of Computer Science of the University of North Carolina at Chapel Hill (UNC), as an assistant professor (2002–2004)
- Department of Computer Science of the University of Crete and the Institute of Computer Science of the Foundation for Research and Technology-Hellas (FORTH-ICS), as an assistant professor (2004–)

Parts of this book are based on research results obtained in joint work with Haipeng Shen, Merkourios Karaliopoulos, Félix Hernández-Campos, George Tzagkarakis, Panagiotis Tsakalides, Elias Raftopoulos, Manolis Ploumidis, Manolis Spanakis, Mark Lindsey, Francisco Chinchilla, Thomas Karagianis, Charalampos Fretzagias, Niko Kotilainen, Lito Kriara, and Konstantinos Vandikas. Haipeng Shen, Merkourios Karaliopoulos, and Félix Hernández-Campos played an important role in the wireless measurement and modeling research. I am grateful to have the opportunity to closely collaborate with them.

Thanks go to Jim Gogan, Todd Lane, Kevin Jeffay, Don Smith and their students for helping to setup the monitoring and data collection system for our network measurement research while at UNC. I would really like to thank all my colleagues at UNC for their help in several different ways. I am also grateful to Diane and Mark Pozefsky for their support while at UNC. FORTH-ICS has provided a state-of-the-art infrastructure to continue my research. Thanks to all my colleagues at the University of Crete and FORTH-ICS for their support that made the transition from US to Greece easier. I would like to acknowledge the support of the director of FORTH-ICS, Constantine Stephanidis and of my colleagues Grigoris Antoniou, Angelos Bilas, Vassilis Christophides, Prasun Dewan, Kevin Jeffay, Manolis Katevenis, Jasleen Kaur, Evangelos Markatos, Christos Nikolaou, George Polyzos, K. K. Ramakrishnan, Montek Singh, Don Smith, Panos Tsakalides, Leandra Vicci, and Gregory Welch. I am also grateful to Anthony Ephremides, Anargyros Papageorgiou, Leandros Tassioulas, and Apostolos Traganitis for their mentoring. Frederick Brooks, Jr. has been a source of inspiration.

Elias Raftopoulos and Manolis Ploumidis—my first graduate students at the University of Crete and FORTH—have been enthusiastically participating in the wireless measurement project. Several other students also contributed in the implementation of CLS, 7DS and applications that use the peer-to-peer paradigm: Denis Abramov and Stelios Sidiroglou-Douskos at Columbia University; Mark Lindsey, Daniel Plaisted, Julien Jomier at the University of North Carolina at Chapel Hill; Niko Kotilainen at Jyväskylä University; and Kostantinos Vandikas, Lito Kriara and Sofia Nikitaki at the University of Crete. It was a pleasure working with all of them.

I am grateful to my editorial assistant Anthony Griffin for reviewing this manuscript several times and providing useful feedback. Several people reviewed the monograph and provided feedback: Thanasis Mouchtaris, Anargyros Papageorgiou, Haipeng Shen, Leandros Tassioulas, Apostolos Traganitis, Panos Tsakalides, and George Tzagkarakis. I would like to acknowledge Antonis Makrogiannakis and Elias Raftopoulos for helping me with *Latex*, and Mary-Rose James and Vana Manasiadi for additional editorial suggestions. I also wish to thank my literary agents Susan Lagerstrom-Fife and Sharon Palleschi for their patience.

Finally, I would like to gratefully acknowledge the support of several agencies:

- the Greek General Secretariat for Research and Technology (Regional of Crete Crete-Wise and 05NON-EU-238)
- the European Commission (MIRG-CT-2005-029186)
- the Department of Computer Science of the University of North Carolina at Chapel Hill for their generous startup fund and the UNC Junior Faculty Development Award
- the Institute of Computer Science of the Foundation for Research and Technology-Hellas, for their state-of-the-art infrastructure and generous startup fund
- IBM, for the IBM Faculty Awards in 2003 and 2004

This monograph also marks an academic journey, that started and ended in Crete. A major visit in this journey: Columbia University and New York City, places that offered intellectual stimulations with such generosity. In 2002, I arrived in a very warm and supportive academic family: the Department of Computer Science at the University of North Carolina at Chapel Hill. Being an assistant professor in this institution—my first real job—was a particularly rewarding experience. My return as a faculty at the Department of Computer Science in the University of Crete—where I had completed my undergraduate studies—offered a sense of continuity that has a strong impact on me.

The following people made an immense impact on enabling this journey:

- Manolis Maragkakis, a beloved math teacher.
- Stelios Orphanoudakis—Professor of the Department of Computer Science at the University of Crete—a charismatic human being that made a large impact on the development of FORTH-ICS and FORTHnet S.A.
- George Papadopoulos and Xacousti Papadopouli-Plevraki, my parents, and Eva Papadopouli, my sister, for always being ready to help, advise, care, and love.

This monograph is dedicated to them.

Maria Papadopouli

Contents

1	Introduction	1
1.1	Wireless data communications	1
1.2	Mobile information access	3
1.2.1	Wireless Internet via APs	5
1.2.2	Infostations	6
1.2.3	Peer-to-Peer systems	7
1.3	Target mobile computing environment	10
1.3.1	High spatial locality of information and queries	11
1.3.2	Heterogeneity in application requirements	11
1.3.3	Enhancement of information access	12
1.4	Resource sharing using 7DS	13
1.5	Overview of this monograph	14
1.5.1	Outline	16
2	7DS architecture for information sharing	19
2.1	Overview of 7DS architecture	19
2.1.1	Communication	22
2.1.2	Cache management	22
2.1.3	Power conservation	23
2.2	Preventing denial-of-service attacks	25
2.3	Encouraging cooperation	26
2.3.1	Micropayment mechanisms	27
2.3.2	Reputation mechanisms	30
2.4	Location-sensing using the peer-to-peer paradigm	30
2.4.1	Overview of CLS	32
2.4.2	Particle filter-based framework	36
2.4.3	Performance of CLS and other related systems	37
2.5	Applications using information sharing via 7DS	39
2.5.1	Web browsing	41
2.5.2	Notesharing and whiteboard tool	41
2.5.3	Multimedia traveling journal	43

2.6	Related mobile peer-to-peer computing systems	45
2.7	Conclusions	46
3	Performance analysis of information discovery and dissemination	49
3.1	Information discovery schemes	49
3.2	Simulation assumptions	52
3.3	Data dissemination benchmarks	54
3.4	Density of dataholders	54
3.5	Average delay	59
3.6	Scaling properties of data dissemination	64
3.7	Models of information dissemination	66
3.7.1	Simple epidemic model	68
3.7.2	Diffusion-controlled process	69
3.8	Discussion	71
4	Empirically-based measurements on wireless demand	77
4.1	Introduction	77
4.2	Campus-wide wireless infrastructure	78
4.3	Monitoring and data acquisition	79
4.3.1	Packet header traces	80
4.3.2	HTTP traces	81
4.3.3	SNMP traces	81
4.3.4	SYSLOG traces	82
4.3.5	Privacy assurances	82
4.3.6	Client identification	82
4.4	State, history, visits and sessions	83
4.5	Wireless traffic demand at APs	84
4.5.1	Data acquisition	84
4.5.2	Comparative analysis of wireless traffic load at APs	87
4.6	Application-based characterization of wireless demand	88
4.7	Locality of web objects	94
4.7.1	HTTP requests model	96
4.7.2	Same-client repeated requests	96
4.7.3	Same-AP repeated requests	97
4.7.4	AP-coresident-client repeated requests	97
4.7.5	Same-building and campus-wide repeated requests	99
4.8	Discussion	101
5	Modeling the wireless user demand	105
5.1	Introduction	105
5.2	Client access patterns	109
5.2.1	Session duration	111
5.2.2	Transient sessions	114
5.2.3	Revisits	115

5.3	Roaming across APs	118
5.4	Arrivals of wireless clients at APs	122
5.4.1	Time-varying Poisson process	123
5.4.2	Arrival process of visits at wireless hotspots	125
5.5	Methodology for modeling user demand	129
5.5.1	Sessions and flows	129
5.5.2	Models of user demand	130
5.6	Syntrig: a synthetic traffic generator	134
5.7	Scalability and reusability in user demand models	137
5.7.1	Variation of the session arrival rate within a day	140
5.7.2	Variation of the session-level flow-related variables	140
5.8	Evaluation of user demand models	142
5.8.1	Statistical-based evaluation	142
5.8.2	Systems-based evaluation	144
5.9	Singular spectrum analysis of traffic at APs	154
5.10	Related work	156
5.11	Conclusions	159
6	Conclusions and future work	161
6.1	Conclusions	161
6.1.1	Mobile peer-to-peer computing	161
6.1.2	Wireless measurements and modeling	163
6.2	Directions for future research	166
6.2.1	Increasing capacity	167
6.2.2	Capacity planning	168
6.2.3	Network interface and channel selection	169
6.2.4	Monitoring	170
6.3	Bio-inspired computing networks	172
6.4	New horizons in cross-disciplinary research	174
	Appendices	175
A	Appendix	177
B	Wireless measurement-based data repositories	179
	References	183
	Index	207

Introduction

“This world!
This small world the great!”
Odysseus Elytis

1.1 Wireless data communications

In the 19th century, the advent of the telegraph and telephone forever changed how messages were transmitted around the world. Radio, television, computers, and the Internet further revolutionized communication in the 20th century. Equally important, the effect of Moore’s law¹ is transforming a niche technology into a ubiquitous one, expanding the innovations in an increasingly networked world. Wireless devices are becoming smaller, easier to use and pervasive. In effect, people are depending more and more on wireless information wherever they are. At the dawn of the 21st century, pervasive computing weaves itself into our lives [352, 6, 4, 29, 42, 48, 23, 50, 47, 38, 19, 22, 18].

Today people access local and international news, traffic or weather reports, sports, maps, guide books, music, video files and games via the Internet [27, 52]. Data volume—medical data, personal multimedia, surveillance for urban areas, web data—is exploding. Similarly, the importance of meta-data, i.e., semantic annotations of what this data means, is also rapidly growing. Analysts expect the growth in mobile location-based services in the European market to reach 622 million euros in 2010, estimating that 18 million users in Europe will subscribe to location-based billing plans by then. Similarly, there is a growing interest in the transportation industry to equip vehicles with navigation tools and location-based services [31, 27, 35, 32, 16, 25]; in the medical

¹ Historically, according to Moore’s Law (posited by Intel founder Gordon Moore in 1965), the number of transistors on a chip roughly doubles every two years, resulting in more features, increased performance and decreased cost per transistor.

community with patient monitoring and assistive technology [9, 7, 28, 21]; and in the entertainment industry, environmental activities and emergency situations for disaster relief. While in 2004, approximately five million portable navigation devices were being shipped worldwide, in 2006 this number was almost quadrupled and it has been forecasted to reach 80 million in 2010. There is also a large increase in the number of PDAs and smartphones world-wide (Table 1.1 and Figure 1.1).

Examples of vehicle-based services are location tracking, maps, driving directions, driver or trip task lists, address lookup, traffic and routing information, fleet tracking, and inter-vehicle entertainment [32, 35]. Within Germany, more than 4,000 motorway sensors nationwide gather data to inform motorists of relevant developments as they happen. Location-aware services have been deployed to provide information about over 1.5 million locations in North America, including hospitals, hotels, banks, ATMs, golf courses, museums, schools, shopping centers, and tourist attractions [32].

In environmental activities, sensors monitor light, temperature, humidity, pollution, barometric pressure, and the presence of animals, reporting data that are typically relayed to central points for analysis and interpretation. Such mechanisms allow biologists to observe and protect habitat with minimal human interference. Entertainment industry uses include mobile gaming, communication and social networking, such as “friends finder” services, posting messages to a map and deciding who can read it, creating and swapping location-tagged photos [53, 52]. The growth of wireless data communications has amplified this trend by making information easier to share, and thus, increasing the amount of information that is shared.

The use of WiFi routers is becoming close to mainstream in the US and Europe. In 2006, 8.4% and 7.9% of all such respective households have deployed such routers and in that year only 200 million chipsets were shipped worldwide, nearly half of the 500 million cumulative total [1]. China already has the same number of mobile-phone users (500 million) as the whole of Europe.

Popular applications and services from wired networks shift to the wireless arena and new applications are increasingly being deployed. The proportion of wireless streaming audio and video traffic increased by 405% between 2001 and 2003/2004, peer-to-peer from 5.2% in 2001 to 19.3% in 2003/4, filesystems from 5.3% to 21.5%, and streaming from 0.9% to 4.6%. Between January 2006 and March 2006, Verizon wireless customers exchanged more than 171 million picture and video messages over its nationwide network.

New applications and tools for storing and sharing information, such as Flickr, YouTube, and Me.dium, have allowed the formation of new types of social networks and online communities.

The value of the networking environments is growing as fast as the number of its users. However, as transistors continue to shrink, running at higher speeds, power consumption and heat become potential limiting factors. More importantly, the demand for information and power is accelerating with the

advancement of displays, graphics, and antennae, and the increase in bandwidth capacity. Although there are improvements in energy consumption, battery capacity grows slowly and power remains an important challenge in mobile computing [46]. Furthermore, as the wireless demand grows, more possibilities for single point failures and service degradation exist. Current wireless devices experience frequent disconnections, packet losses, and delays, while wireless infrastructures are unable to successfully support applications with real-time constraints. A denser deployment of wireless networks may alleviate the problem of intermittent connectivity but would exacerbate the interference, if carried out indiscriminately.

Two distinct aspects of wireless communication that make wireless networks more vulnerable than the wired ones are the fading and the interference between receiver and transceiver. The phenomenon of fading is characterized by the time variation of the channel strengths due to the small-scale effect of multipath fading or the larger-scale effects due to attenuation and shadowing by obstacles. Examples of various wireless technologies with their bandwidth requirements, frequency, and effective range are presented in Table 1.2.

	2001	2002	2003	2004	2006	2008
PDA	15,336	15,714	18,946	23,854	38,320	58,509
Phone-PDA	4.3	10.8	20.6	29.3	39.9	45

Table 1.1. PDAs and smartphones worldwide (thousands). Source: eTForecasts report on “Worldwide PDA Markets” [14]. Handset makers: 1.14 billion sold in 2007, 987 million sold in 2006, 1.43 billion handsets to be sold in 2011 (Source: FT, Gartner). The number of subscribers using mobile Internet services will rise from 577 million currently, to top 1.7 billion by 2013.

1.2 Mobile information access

Mobile information access is the underlying querying and data acquisition mechanisms via which a wireless device searches for, and receives information from other devices while mobile. The mechanism describes the system architecture, its main components, and its interactivity model. The latter characterizes whether or not the communication between the “data-querier” and “data-provider” is synchronous. In synchronous access, a *user* specifies a data request in real-time, and the system accesses the information from the source or its local cache. Thus, a dependency between the request for data and the corresponding response exists. Alternatively, in the asynchronous case, the request is triggered by an event or an application and the system does not wait for a response.

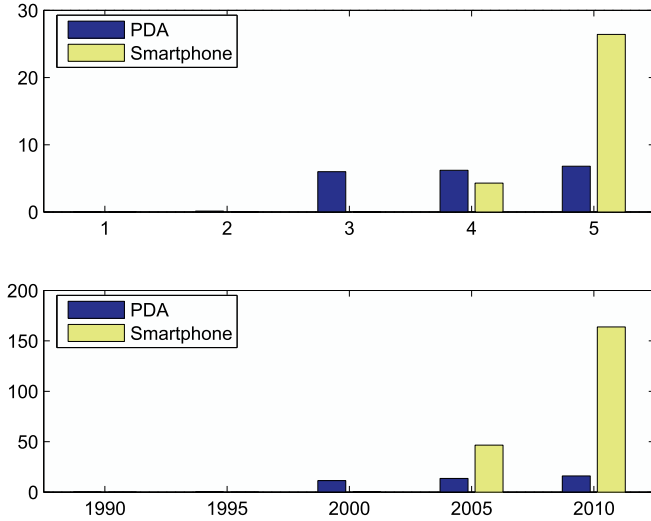


Fig. 1.1. The market growth in handheld devices (in millions).

Technology	Maximum bit-rate	Frequency	Effective range
Bluetooth	724 Kbps	2.4 GHz	10 m
Infrared	<4 Mbps	$> 10^5$ GHz	20 m 100 m 10 cm – 2 m
IEEE802.11b	1 Mbps 11 Mbps	2.4 GHz	outdoors 550 m indoors 50 m outdoors 160 m indoors 50 m
3G	144 Kbps vehicle 384 Kbps pedestrian 1-2 Mbps stationary	1.885 -2.2 GHz	50 km
CDPD	19.2 Kbps	1.8-2.5 GHz	

Table 1.2. Examples of various wireless technologies with their bandwidth requirements, frequency, and effective range.

Prefetching or *hoarding* is a type of asynchronous access in which, prior to its disconnection, a device prefetches the data from the file system. It aims to alleviate user-perceived latencies by providing data while the device remains disconnected and reintegrating upon reconnection [226, 242]. Hoarding

strategies exploit the detection of “file working sets” [337] and semantic relationships among files. Designed for traditional file-systems settings, hoarding is appropriate when the system can predict and locate the information to be prefetched. However, it can be inadequate in dynamic environments when a device searches for new data while mobile.

The mobile information access can be classified according to its dependency on an infrastructure and interactivity model into the following three main categories, the first two of which require an infrastructure:

1. *wireless Internet via APs*
2. *data access via infostations*
3. *data access using the peer-to-peer paradigm*

1.2.1 Wireless Internet via APs

A wireless *access point* (AP) is a device that connects other wireless-enabled devices in its wireless range to form a wireless network. Usually, it connects to a wired network, and can relay data between wireless-enabled devices in its range and devices of the wired network. Within the range of an AP, a wireless end-user has a full network connection with the benefit of mobility. Many APs can be connected together to create larger networks that allow “roaming” between them; APs relay packets between each other, so that a packet can be delivered to its final destination, a roaming client. In contrast to infrastructure-based networks, *ad-hoc networks* operate in a self-organizing, autonomous manner.

APs may also form *mesh networks*. In general, mesh networks are ad-hoc multi-hop networks with a mesh topology, that consist of mostly stationary wireless devices that cooperate with one another to route packets, forming the network’s backbone. In addition to the routing capability for gateway/bridge functions as in a conventional wireless network, these *mesh routers* support routing mechanisms for mesh networking. With gateway functionality, mesh routers can be connected to the Internet. Non-routing mobile devices or *mesh clients* can connect to mesh nodes and use the backbone to communicate with one another over large distances and with nodes on the Internet. Clients with an Ethernet interface can be connected to mesh routers via Ethernet links. Thus, mesh networks are heterogeneous, hybrid and possibly multi-operated networks, composed of wired and wireless, stationary and mobile devices. Unlike mesh routers that may not have power constraints, typical mobile clients require the support of power-efficient mechanisms.

Mesh networks extend high-speed local area networking services to a wider area. A number of community wireless mesh networks exist, such as the Seattle Wireless and Roofnet networks. The latter is a 38-node multi-hop IEEE802.11 network spread over four square kilometers of an urban area. Commercial mesh Internet access services and technologies include MeshNetworks Inc., Ricochet, Meraki Networks, and Tropos Networks.

The wireless Internet via APs aims at “continuous” wireless Internet access broadly defined by three types of wireless networks, namely, wireless wide area networks (WANs), wireless local area networks (LANs), and wireless personal area networks (PANs). Examples include: CDPD, 3G wireless, IEEE802.11 and two-way pagers [137, 299]. Table 1.3 presents some examples of U.S. wireless networks and their wireless transmission technology.

Technology	Carrier
TDMA	AT&T (Cingular), Digital PCS, CellularOne
GSM/GPRS	Omnipoint, AT&T(Cingular), Voicestream, Unicel, PinPoint Wireless
CDMA	AirTouch, Verizon, General Wireless, Sprint PCS, MCIWorldCom
	Qwest, Bell Atlantic Mobile
CDPD	Digital PCS, BellAtlantic/Nynex, AT&T Verizon Wireless, Omniskey

Table 1.3. Examples of U.S. wireless networks and their wireless transmission technology.

Wireless WANs are licensed, strictly regulated wireless networks used by cell phones and wireless modems; examples include CDPD, TDMA, GPRS, GSM, 3G wireless, and two-way pagers. Wireless WAN access is typically characterized by low bit-rates and long delays. Unlike wireless WANs, wireless LANs, such as IEEE802.11, HiPERLAN, DECT, operate in unlicensed spectrum.

In several cities worldwide, nonprofit, educational, and commercial organizations have installed IEEE802.11 APs to provide free wireless access to the Internet (e.g., Figure 1.2). In the late 1990s and early 2000s, APs grew rapidly in popularity, as they were low-cost and simple mechanisms to expand the wireless connectivity of an existing infrastructure.

Wireless PANs are short-range, low-power networks via Bluetooth, HomeRF, RFID, IrDA, and IEEE802.15 technologies. Such networks are already deployed in home and office environments.

These new technologies and uses raise new issues related to ethics, security, privacy, confidentiality, and legislation. Take as an example the RFID tagging: While there are interoperability issues, researchers predict that within a twenty-year period, RF tags will be pervasive, first as passports, driver’s licenses, medical bracelets, credit cards, and then, as implantable chips in humans. Even more data will be captured, stored, and analyzed. Implanting RF tags in humans provokes numerous ethical, legislation, and privacy-related concerns [163, 15].

1.2.2 Infostations

An infostation is a wireless-enabled server attached to a data repository. Wireless devices in the range of an infostation can query the infostation to acquire data. Although typical infostations are stationary, we can envision robots

roaming an area and acting as mobile infostations. Like APs, infostations can be stand-alone servers or clustered with other infostations and connected over terrestrial links, such as T1, SONET, and/or fiber.

An infostation located in popular areas—such as at traffic lights, building entrances, cafes, and airport lounges—can provide information access to users in their short-range, operating according to the server-client paradigm.

In general, a client can acquire the data from an infostation in an asynchronous or synchronous manner. For instance, an infostation may multicast the data periodically, while clients subscribe to this multicast channel to receive the relevant information. The infostation paradigm can be extended to a network of infostations that act as proxies, caching data and forwarding requests to other infostations or to the Internet. Infostations were first mentioned by Imielinski and Badrinath in the DataMan project [303, 198].

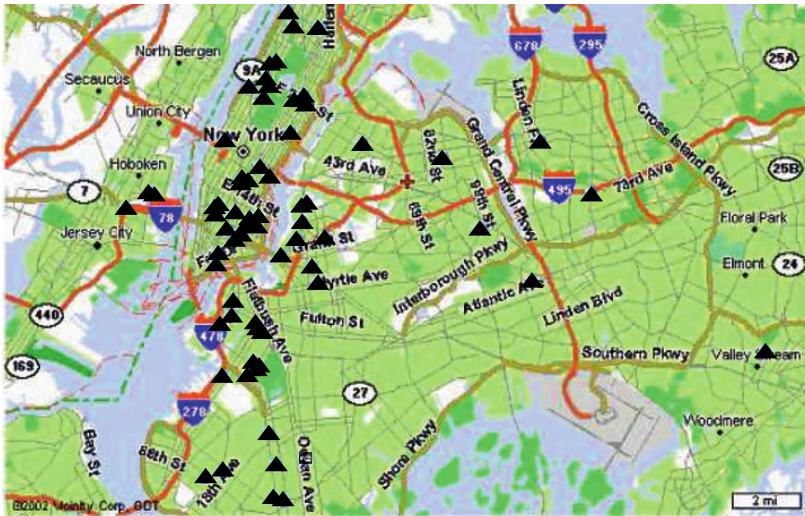


Fig. 1.2. The New York City wireless public access points as of May 2002 [34]. The wireless access points are depicted as solid triangles.

1.2.3 Peer-to-Peer systems

A *peer-to-peer* system is a *distributed system without any centralized control or infrastructure*. The software running at each peer host is *equivalent in functionality*, so that peers can dynamically share their resources by both requesting and offering services, rather than being confined to either client or server roles. Peer-to-peer systems are distinguished by the following main criteria:

- *self-organization*
- *autonomy*
- *symmetry*

The peer-to-peer paradigm does not require the support of any infrastructure and is based on the resource sharing among wireless devices. These devices (or simply peers) cooperate dynamically based on some policies that specify their cooperation and functionality. Unlike the traditional client-server model, in peer-to-peer computing, there is no centralized powerful device or cluster of devices and participants (peers) communicate to discover and share resources. Examples of such resources are computing power, data, and network bandwidth.

The peer-to-peer concept was originally introduced in the context of distributed systems, but in the mid-1980s, the term was used by local area network vendors to describe their connectivity architecture.

The term reappeared in 1999 with the widespread popularity of Napster [272] and by early 2001, Napster claimed over 60 million registered users sharing terabytes of music files. Like Napster, Gnutella [159] and Freenet [49] are two other peer-to-peer systems that gained popularity in early 2000s by enabling users to share data in a fixed wired network. While Napster had focused on sharing music files and Gnutella any type of file sharing, Freenet facilitated encrypted and anonymized distributed storage.

In early peer-to-peer systems, such as Gnutella and Freenet, peers were “blindly” sending their requests to many other peers without keeping track of which peer had a specific document, resulting in large searching delays. Later, peer-to-peer systems, such as CAN and Chord [334], imposed a consistent mapping between an object key and a peer in the network. Each peer maintains information about a number of other peers in the system, creating a logical topology that provides some guarantees about searching delays.

In the late 1990s, the research community had been investigating replicated storage systems based on the peer-to-peer architecture meant for wide-scale, Internet-based use. Examples of these research efforts include the Ficus [281], JetFile [165] and Bayou [332], with main focus on update policies, data consistency, and reconciliation algorithms. Since then, research in peer-to-peer systems has considered mostly wired-based infrastructure and use, aiming to improve scalability, robustness, and efficiency in routing, indexing, and information searching and dissemination.

Skype is a popular Internet telephony program that applies the peer-to-peer paradigm. The peer-to-peer paradigm has been also utilized for content distribution, such as OS or anti-virus updates, in a wired-based infrastructure of PCs. Examples of such systems are Limewire [24], OpenFT [36], BitTorrent [199] and Avalanche [158].

BitTorrent has quickly emerged as a viable and popular alternative to file mirroring for the distribution of *large* content [85]. To share a file or group of files through BitTorrent, clients first create a file with meta-data, such

as a description of the files to be shared, the host that coordinates the file distribution, suggested names for the files, their lengths, the piece length used, and a SHA-1 hash code for each piece to be used to verify the integrity of the received data. After the creation of this file, a link to it is placed on a website or elsewhere, and it is registered with a tracker which maintains lists of the current participants. A client that has downloaded a file may also act as a dataholder, providing a complete copy of the file.

The information theory community has proposed some routing protocols in ad-hoc networks based on cooperative diversity schemes. These schemes send information through multiple relays concurrently. The destination can then choose the best of many related packets or combine information from multiple packets to reconstruct the original data. Avalanche, for example, uses network coding techniques that allow each PC in the distribution network to generate and transmit blocks of information. Avalanche peers produce linear combinations of the blocks they have already cached. Such combinations are distributed together with a tag that describes the parameters in the combination. Any peer can generate new unique combinations from the combinations it already has. A peer can decode and build the original file when it has sufficient independent combinations. The network encoding ensures that any block uploaded by a given peer can be of use to any other peer.

Today, in wireless campus infrastructures, the web and peer-to-peer are the most dominant application types both in terms of number of flows and bytes. One of our recent measurement studies [300] showed that around 30% of the flows (or 20% of total bytes) accessed via the wireless infrastructure of the UNC campus in April 2005 had been generated by peer-to-peer applications and around 70% of clients had at least one flow generated by a peer-to-peer application. Additionally, BitTorrent peer-to-peer file sharing was found to be the biggest consumer of bandwidth, accounting for about 30% of the total data transferred in an application-based traffic classification study in the Roofnet mesh network [85].

While web requests accounted for a minority of the data transferred, they contributed a larger number of flows than any other application (68% of the flows were web, compared to the 3% that were BitTorrent).

Since the appearance of wireless ad-hoc networks, the peer-to-peer paradigm has been playing a prominent role in routing protocols for such networks [92, 324, 360, 139, 326, 297, 157, 194, 218, 192]. Typical ad-hoc networks assume cooperative devices that will relay a packet until it reaches its final destination in *dense, large-scale, mostly-connected* wireless networks. More recently, mesh networks have been instantiating the peer-to-peer paradigm with their “grass-roots” approach to provide wireless access with a minimum infrastructure that creates a mostly stationary multi-hop network. Sensor networks—often composed by devices, unattended, with limited capabilities—may form ad-hoc networks for monitoring various environmental conditions. There are two clear trends in the networking horizon: more and more networks become

from centralized, to distributed, to autonomous, self-organized and pervasive. Devices become smaller, more networked and more programmable.

Another manifestation of the mobile peer-to-peer paradigm has taken place in rural areas of developing nations with vehicles offering web content to computers with no Internet connection. Specifically, the United Villages project [344] provides villagers in Asia, Africa, and Latin America with a digital identity and access to locally-relevant products and services using a store-and-forward, “driven-by WiFi” technology. The mobile APs are installed on existing vehicles (e.g., buses and motorcycles) and automatically provide access along the road. Whenever a mobile AP is within range of a real-time wireless Internet connection, it transfers the data from and for those kiosks.

In this work, our attention shifts to wireless networks that are *sparser* and *frequently disconnected* from the Internet. In such networks, a device is not always connected to the Internet, nor within wireless range of another device. Real-life networks exhibit a large diversity in application requirements, device characteristics, connectivity, density and cooperation, and scale.

1.3 Target mobile computing environment

Environments that exhibit the following two characteristics particularly motivated this research:

- *frequent disconnections* from the wireless Internet due to mobility
- *high spatial locality of information*

A network of wireless devices is characterized by high spatial locality of information when wireless devices in close geographic proximity access similar data. For example, devices running location-based services that are in close proximity request similar type of data, such as traffic reports, and popular tourist sites.

This networking environment may encompass a wide range of wireless-enabled devices with different energy and storage constraints, various network interfaces, mobility patterns, and incentives for cooperation with each other. It may include handheld devices (such as iPAQs, palm pilots, and mobile phones) with memory and power constraints, devices with higher availability in storage and power (such as laptops or vehicular wireless-enabled systems), and infostations with sufficient storage and no power constraints. Devices may be autonomous, not necessarily connected to the Internet, mobile or stationary.

Currently, mobile users access information using a wireless LAN or WAN infrastructure. Most wireless data WAN access, such as Vindigo [349] or RIM [315], is only available in major metropolitan areas. Although IEEE802.11 networks have become widely available in universities, corporations, and public areas providing wireless LAN access; areas abound in which communication infrastructure is either not available or overloaded, and expensive to access.

Examples are: in emergency situations, disaster relief, rescue operations, inside a tunnel or in a rural area.

Given the exorbitant license fees paid out in recent government auctions of spectrum, the bandwidth expansion route is bound to be expensive. For example, European telecommunications giants spent \$100 billion in 2000 for 3G license fees [164]. Similarly, the cost of tessellating a coverage area with a sufficient number of APs or infostations coupled with the cost of associated high speed wired infrastructure may be prohibitive. Though conditions vary widely, building underground fiber networks in highly congested urban areas can cost \$100 or more per foot of cable installed. In contrast, placing fiber underground in the suburbs costs \$7 to \$25 a foot. More importantly, the deployment of APs without capacity planning or mechanisms for dynamic AP-configuration and self-organization—in terms of power control, channel selection, user admission control and bit-rate selection—may result in interference and degradation of the wireless access.

For the next few years, continuous connectivity to the Internet world-wide will not be available at low cost for mobile users roaming a metropolitan area; devices will continue to experience changes in the availability of bandwidth and frequent interruptions of connectivity due to host mobility.

1.3.1 High spatial locality of information and queries

The growing popularity of location-dependent services, collaborative applications, peer-to-peer systems, and interactive games running on mobile devices will result in high spatial locality of information. For instance, in an urban environment, an airport, or a commercial center, users with wireless-enabled devices access local and world news, sports news, train schedules, weather reports, maps, and routes. Similarly, users in a corporation, in an academic department, or at a gathering, may share photos or video clips from their recent vacation; while people standing in the line of a theater, or in front of a sculpture in a museum, may share reviews about the play or exhibition.

An increasing number of wireless Internet and information providers target handheld devices, e.g., Avantgo (Figure 1.3), Vindigo, Omnisky Corp.. For example, Avantgo regularly listed *The Wall Street Journal*, *The New York Times*, and *USA Today* as the top ten user sites at www.avantgo.com/channels. Similarly, Vindigo licenses its technology to newspapers and hosts the service on behalf of its partners. Newspapers simply supply the listings in a structured format and update them periodically. In a different networking environment, a highway, vehicles with wireless access request weather and traffic reports, maps, and routes, generating queries with high spatial locality information.

1.3.2 Heterogeneity in application requirements

Applications dictate particular requirements for reliability, delay, and bandwidth that can vary greatly. Unlike voice communication, many wireless applications are delay-tolerant, i.e., they possess loose delay constraints (of the

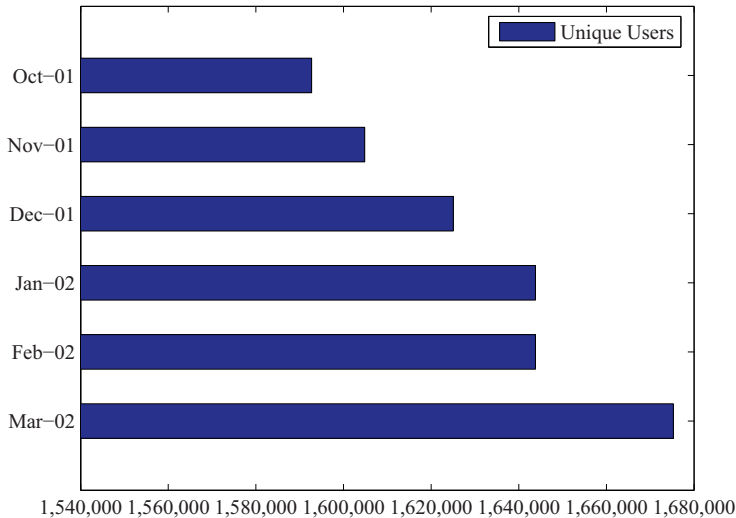


Fig. 1.3. The unique number of Avantgo users that subscribe to the NYTimes news on-line information, respectively. Source: *The New York Times on the Web* [279].

order of minutes). In pervasive computing, context-based information may change dynamically and can be inherently imprecise. Depending on the application, users may have flexible requirements regarding information accuracy, freshness, precision, and media quality. Often users may trade the response time for less timely or lower resolution data. In other cases, up to a few hours of delay can be tolerated, as long as messages eventually reach their destination (e.g., tourists with wireless-enabled cameras that wish to send photographs home).

1.3.3 Enhancement of information access

As discussed previously, mobile information access via an infrastructure of APs or infostations exhibits frequent disconnections and low bit-rates. Our main challenge is to provide *complementary* mechanisms that enhance the information access when mobile devices face disconnections to the Internet. To achieve this, we proposed a mobile peer-to-peer computing paradigm that enables resource sharing when an infrastructure is not always available. This paradigm was also analyzed, evaluated and compared with more traditional mobile access methods, namely, via APs and infostations.

1.4 Resource sharing using 7DS

We propose 7DS, an architecture and set of protocols that enable *resource sharing among peers* that are *not necessarily connected* to the Internet. 7DS encompasses three facets of cooperation: *data sharing*, *message relaying*, and *bandwidth sharing*.

7DS may relay, search for and disseminate information, and share bandwidth. It operates in a self-organizing manner, without the need for an infrastructure and serves as the underlying information and service discovery protocol. We assume that 7DS runs in the middleware and 7DS-enabled devices communicate with each other via wireless LANs.

7DS stands for “Seven Degrees of Separation”, a variation of the “Six Degrees of Separation” hypothesis, which states that any person can be connected to any other person through a chain of acquaintances with no more than five intermediaries. An analogy to our system can be made, particularly with respect to data recipients and the device with the original copy. The six degrees of separation was a popularized version of the small world concept, a term coined by the sociologist Stanley Milgram in the 1960s in the context of his experiments on the structure of social networks². 7DS was inspired by the idea that there will be a growing number of “on-line” communities of mobile users that gossip, share information and resources via their wireless-enabled devices.

7DS-enabled devices can interact either in a *peer-to-peer (P-P)* or *server-to-client (S-C)* manner. The S-C mode is asymmetric; there are 7DS-enabled servers that respond to queries and non-cooperative, potentially resource-constrained clients.

Throughout the text, the term *7DS node* or *7DS host* or simply *host* are used interchangeably to indicate any 7DS-enabled device, and *7DS peer* or simply *peer* any 7DS-enabled device that employs the peer-to-peer paradigm. These different modes of operation allow 7DS to instantiate different mobile information access schemes when possible, and provide complementary access through peers, when an infrastructure is not available.

Hosts can be handheld devices that are mobile and power constrained, stationary PCs, and servers or infostations connected to the Internet and a power outlet. A 7DS-enabled server can be either a dual-homed device connected to the Internet or a wired infrastructure of other servers, or an autonomous infostation. It can be mobile or stationary. An example of mobile server is a robot that roams in a museum and disseminates information to visitors with handheld devices. 7DS running on handheld devices will use different power conservation and collaboration methods than 7DS-enabled servers.

7DS nodes can collaborate by data sharing, forwarding messages or caching popular data objects. For example, an autonomous 7DS server may monitor for frequently requested data, request it from other peers, and store it locally

² We have not explored if a similar hypothesis is true here.

to serve future queries. The fixed information server (FIS) is an instantiation of the S-C scheme with a stationary server and is equivalent to the infostation model. Thus, 7DS can be viewed as a generalization of the infostation concept.

In information sharing, peers query, discover, and disseminate information. A 7DS host acquires data from other peers (in P-P) or from the infostation (in S-C) within its wireless coverage using single-hop broadcast to periodically query for data. Instead of operating with high transmission power to reach an AP or an infostation that is far away, a host forwards its messages or requests for data to its peers in close proximity. In that way, hosts can conserve more power and better utilize wireless bandwidth. Replication introduces a tradeoff among data consistency, security vulnerabilities, management overhead, and availability. 7DS assumes that, in the face of disconnections, users can trade the data consistency and currency over data availability.

Motivated by the high spatial locality, which is intrinsic in positioning, we also applied the peer-to-peer concept to location-sensing. Specifically, we designed a collaborative location-sensing system (CLS) that adaptively positions wireless-enabled devices using the *existing* communication infrastructure and without the need of specialized hardware. CLS enables hosts to cooperate by sharing their position estimates, and use these estimates along with signal strength measurements, to iteratively determine their position.

To conserve power, 7DS periodically activates the network interface. During the *on* interval, 7DS hosts communicate with their peers. In its asynchronous mode, the *on* and *off* intervals are equal but not synchronized, while in synchronous mode, the *on* and *off* intervals are synchronized among hosts but not necessarily equal.

When *bandwidth sharing* is enabled, 7DS allows a host to act as an application-layer *gateway* and share its connection to the Internet with other hosts. When a peer is unable to access the Internet, it may ask other peers to act as gateways. Alternatively, hosts can buffer their messages locally and relay them to peers. Specifically, in *message relaying*, a host forwards its queued messages to another peer or AP. To prevent message looping and better utilize the buffer, 7DS may restrict the number of times that a message is forwarded and delete old and duplicate ones.

1.5 Overview of this monograph

This work explores two main research domains:

- mobile peer-to-peer computing
- wireless networking measurements and modeling

Its first part presents a novel framework for mobile wireless data access based on the peer-to-peer paradigm. Unlike typical peer-to-peer approaches in wired networks, 7DS does not try to establish permanent caching or service discovery mechanisms due to the highly dynamic environment. Instead, 7DS hosts

acquire the data from other peers within their wireless coverage using single-hop broadcast. The thrust of this research is the information dissemination in mobile networks, which raises several questions: How fast does information spread in such networks? What is the impact of cooperation, data popularity, and wireless range on information diffusion? How do the different mobile information access and caching paradigms compare? The peer-to-peer paradigm is then applied in location-sensing and its performance is evaluated. Given the dearth of large-scale, non-controlled 7DS-like environments, we run extensive simulations to study these issues. We also experiment with novel applications that use 7DS as their underlying information discovery mechanism.

Although IEEE802.11 APs and clients are rapidly deployed, there are still areas of limited or no wireless coverage. 7DS can “bridge” the access via wireless infrastructures and peers through caching and relaying.

To uncover the weaknesses and distinct characteristics of wireless networks, measurement-based studies are critical. Eager to better understand the characteristics of wireless access and workload, we perform empirical analysis and modeling studies. For this purpose, extensive real-traces were acquired from a large-scale campus-wide IEEE802.11-based infrastructure. Their prevalence impels us to analyze them and examine the spatial locality of the wireless information, access patterns, and workload characteristics. Several questions stimulate this research effort: How loaded are the APs and what type of applications are accessed? What is the impact of different caching paradigms in wireless networks? How do users arrive at APs? How do they roam across APs? What are the right structures to model the user-initiated activity in a wireless network?

Most of the performance analysis studies of wireless networking protocols employ as input for the traffic demand traces based on various constant-bit-rate UDP and TCP flows or “infinite” UDP/TCP sources to simulate asymptotic conditions. We are eager to explore models that can reflect realistic workload conditions and at the same time are simple, flexible, and expressive enough to allow us to “manipulate” them in order to simulate or emulate different conditions with respect to the application mix, roaming pattern, and traffic load. We capture the user-initiated activity through flows, *sessions*, i.e., episodes of continuous wireless access in the infrastructure, and disconnections. Furthermore, we present a methodology for modeling the demand and specifically, the client associations at APs, sessions, and flows.

As more mobile peer-to-peer applications and delay-tolerant networks (DTN) [12] are being deployed, it should be easier to acquire traces from such testbeds and apply the proposed modeling methodology to study their access patterns. Such models can then be imported in performance analysis studies to offer more meaningful insight about the performance of various type of protocols.

To summarize, this work presents the following:

1. The design and implementation of 7DS, a novel system that enables information dissemination and sharing among mobile hosts.
2. The evaluation of the impact of the wireless range, host density, querying mechanism, power conservation, and cooperation on data dissemination via extensive simulations.
3. A discussion on theoretical models for data dissemination that use random walks and diffusion-controlled processes.
4. A brief presentation of CLS, a location-sensing system that employs the peer-to-peer paradigm to enhance the position estimates.
5. A measurement-driven evaluation of the spatial locality property of web requests and caching schemes in a large-scale wireless infrastructure.
6. A measurement-driven analysis and modeling of the access patterns and user workload in large-scale wireless infrastructures.
7. Accurate and scalable models of user workload and a discussion of the scalability and reusability tradeoffs.
8. A performance analysis of a wireless LAN that highlights the impact of various traffic models.

1.5.1 Outline

Chapter 2 gives an overview of the main components of 7DS, CLS, and applications that have been integrated with 7DS. Its main results have also appeared in [151, 220, 345, 293, 231].

Chapter 3 evaluates several mobile information access schemes with extensive simulations and presents some theoretical data diffusion models. Most of the results of Chapter 3 have appeared in [283, 284, 285].

The empirical studies included in this book used extensive traces collected from the wireless infrastructure at UNC. Chapter 4 introduces the wireless infrastructure, monitoring tools, data acquisition process and traces, and lists the type of publicly available wireless traces. The main definitions and concepts for modeling the workload are presented, followed by an analysis and an application-based characterization of the wireless workload. Finally, we also examine the spatio-temporal locality of the web requests accessed from the wireless infrastructure and evaluate several caching paradigms using extensive traces. A detailed discussion of the workload analysis and characterization study can be found in [115, 181, 300].

Chapter 5 discusses our multi-level modeling of the wireless demand, namely, the associations and generated traffic in a large-scale wireless network. It provides an empirical modeling of wireless user access: arrivals at APs and roaming patterns across APs. Specifically, it analyzes the duration of a client association at an AP and the roaming between APs and proposes an algorithm that predicts the next AP for a client. It then shifts the perspective from client- and AP-level to an infrastructure-wide view and models main features of the wireless user activity, namely, the episodes of continuous wireless

access and the flow generated during those episodes. A more detailed description of this research can be found in [115, 288, 182, 289, 179, 217, 343, 342].

Finally, Chapter 6 summarizes our results and discusses directions for future work.

7DS architecture for information sharing

This chapter focuses on the architecture components that enable information sharing via 7DS. Firstly, the communication, cache management, and power conservation are presented, followed by a discussion about mechanisms to stimulate cooperation and prevent denial of service attacks. To support location-based applications, we introduce a positioning system, the Cooperative Location-sensing System (CLS) that also applies the peer-to-peer paradigm via 7DS. This chapter gives an overview of CLS, and shows how 7DS can act as the underlying information discovery mechanism for different location-based and collaborative applications.

2.1 Overview of 7DS architecture

A major contribution of computer science—that has played a dramatic role in society and other sciences—is the creation of new paradigms, technologies, and tools for communication and interaction. The World Wide Web and Internet have been catalysts for the creation of collaborative applications and tools. Powerful drivers for on-line collaborations have been “group-forming networks” that allow users to self-organize and form groups, such as eBay, Wikipedia, and the Open Source Initiative. On-line collaboration has been enriched with new applications and tools for storing, sharing, and experimenting with multimedia data, such as Flickr, YouTube, Me.dium, My Space, facebook, and JumpCut. These technologies have allowed the formation of new types of social networks, interactions, and online communities. The communication paradigms, interaction rules, and network topologies can vary and have a great impact on the performance of the information diffusion. Social network analysis has emerged not only as a popular topic of speculation, but also as a key technique in sociology, anthropology, geography, economics, biology and computer science.

7DS facilitates collaboration of mobile devices by instantiating three main information access methods: via an AP, using an infostation, and applying

the peer-to-peer paradigm. It acts as the underlying information discovery mechanism for applications that run on the local device, enabling the peer-to-peer data sharing when access via an AP or server fails.

The novel aspect of 7DS is its instantiation of the peer-to-peer paradigm in a mobile wireless network. The design and implementation of this aspect will be the focus of this chapter. When an application requests a data object, 7DS first checks its cache, and if the data is not available or has expired, it tries to acquire it from the Internet. For example, in the case of web browsing, a data object is a web page including all its embedded files. If the local web browser fails to connect to the web server, 7DS attempts to acquire the page from another peer in the wireless LAN.

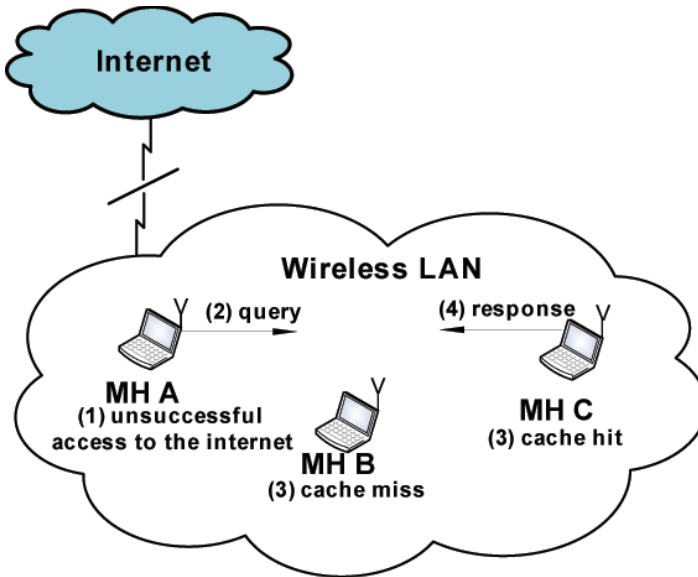


Fig. 2.1. Example of information sharing using 7DS. The arrows show the message exchange for the 7DS communication. The light-shaded area denotes the wireless LAN, the darker-shaded area the Internet, and the thunderbolt-like shape the wireless WAN connection that is not currently available.

Figure 2.1 illustrates an example of 7DS use. Mobile host A (MH A) tries to access a data object. The local 7DS instance running on host A detects an unsuccessful attempt to connect to the Internet and tries to retrieve the page from peers that are within its wireless range. Both hosts B and C (MH B and MH C, respectively) are within the range of host A and receive the query.

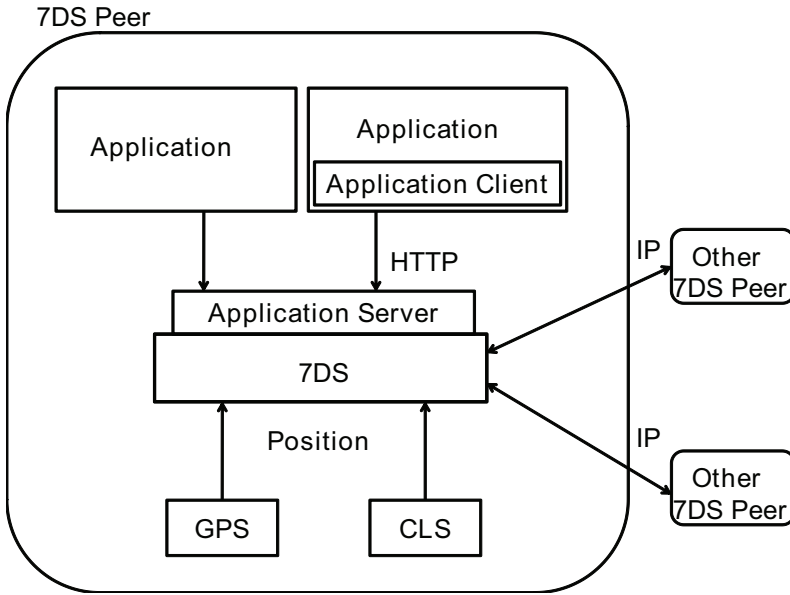


Fig. 2.2. 7DS architecture: an underlying information discovery mechanism for location-based applications, in conjunction with positioning systems (e.g., GPS and CLS).

Unlike host B, host C has a copy of the data in its cache and responds to host A's query.

To facilitate the interaction with 7DS, applications use pairs of attributes, (name, value), to describe the data that they are willing to share with other application instances running on peers. For each application, 7DS maintains an index of the local cache that is populated with data that can be shared. This data may have been acquired from other peers or servers. Figure 2.2 illustrates the general 7DS architecture coupled with positioning systems and applications.

In contrast to Gnutella and other peer-to-peer mechanisms in wired networks, a 7DS peer does not maintain connections with other peers but only multicasts its queries to a well-known multicast group. In addition, 7DS—in the default mode—restricts the query propagation to the wireless LAN. Unlike Napster, 7DS operates in a distributed fashion without the need for a central indexing server. Napster also requires user intervention for uploading files, whereas 7DS does this automatically. Furthermore, our setting is orthogonal to the service discovery in the wide area network. In service discovery, there is typically an infrastructure of cooperative servers that create indices to locate

data based on the queries and the content of the underlying data sources of their local domain [106].

2.1.1 Communication

Applications in a 7DS-enabled system employ *insert* and *query* messages to communicate with their local 7DS instance using SOAP, which is simply XML over HTTP. Specifically, insert messages indicate what data can be shared with other peers and stored in its local cache (Figure 2.3).

The communication among 7DS peers is implemented by the following message types, all in XML format:

- *queries*
- *reports*
- *advertisements*

Queries describe the requested data items with predefined application-specific attributes, and are generated by the application when the relevant data cannot be found locally. In addition, queries include attribute pairs with undefined values to be bound during a matching process at a peer. 7DS supports various types of queries, such as, queries with a list of attributes that must match, *nested boolean* operations, and different types of matches (e.g., case-sensitive exact match, regular expression match).

A 7DS host *actively queries* for a data object when it periodically multicasts a query for that object to a predefined group until it receives the relevant data. Active querying is the default querying mechanism. After receiving a query, a peer extracts the embedded attributes and performs an attribute-matching search in its local cache. In the case of a match, the peer broadcasts a report that describes the relevant data found in its local cache. This generated report reflects the received query, with a subset of its attributes bound via this matching process that is performed locally. A report can be self-sufficient or include a URL for a subsequent retrieval of the complete data object (e.g., Figure 2.4). After a predefined interval, the querying host selects the most relevant report—among the received ones—based on application-specific criteria.

Advertisements are messages periodically multicast from the 7DS-enabled servers to announce their presence. Upon the receipt of such advertisements, a 7DS host may send its queries to the server. As opposed to active querying, this type of querying—defined as *passive querying*—is targeted at power-constrained devices that participate in 7DS only when the requested data is likely to be available.

2.1.2 Cache management

Primary information propagation occurs through the use of caching rather than reliable state maintenance, and 7DS does not attempt to resolve inconsistency among copies of a data object. 7DS organizes and indexes its cache,

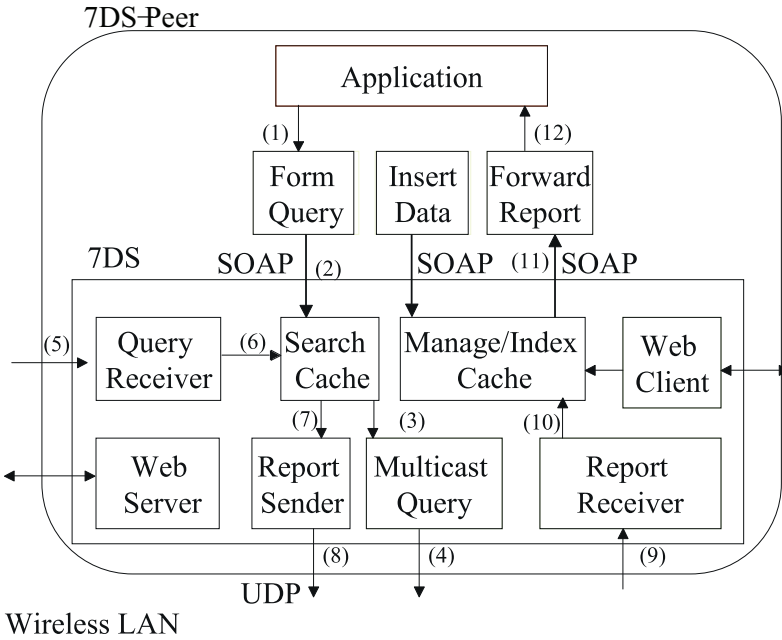


Fig. 2.3. Interaction of 7DS with applications. The communication between 7DS and an application is via SOAP. Only the communication components of 7DS and its interaction with an application are illustrated. The squares inside 7DS indicate the logical modules of 7DS and the arrows the sequence of interaction between them.

which can be viewed, browsed, and managed through a graphical user interface (GUI). The current prototype displays the content of the cache in a directory-like structure (Figure 2.5). The GUI can be extended to support grouping of the cache content by predefined categories and searches using the meta-data attributes of the stored objects. To protect the user's privacy, 7DS only shares reports and pages that correspond to publicly available objects. The cache management includes setting of access permissions of files and directories, deleting expired objects or specific files, and updating the index.

7DS can be easily extended to support the prefetching operation. Through a GUI, users can mark which pages need to be prefetched or updated regularly, and upon their expiration 7DS will generate the corresponding queries.

2.1.3 Power conservation

Using a battery monitor and power management protocol, 7DS aims to adapt its communication pattern to reduce energy consumption, especially when

```

<?xml version="1.0" encoding="UTF-8"?>
<ds:Report xmlns:ds="http://www.cs.unc.edu/~maria/7ds/">
  <ds:Object> <ds:ObjectType>Hypermap</ds:ObjectType>
    <ds:ID>300</ds:ID>
    <ds:SourcePeer>192.168.1.100</ds:SourcePeer>
    <ds:PathToFile>F8F84640FD800549694E2B4C5A6C7198.xml</ds:PathToFile>
    <ds:IsPrivate>false</ds:IsPrivate>
    <ds:Application>
      <Description>SVG Project Demo</Description>
      <Type>Meeting</Type>
      <Start Time>1051905600000</Start Time>
      <SVGMapYCoordinate>-1495.0</SVGMapYCoordinate>
      <EndTime>1051909200000</EndTime>
      <SVGMapID>0</SVGMapID>
      <SVGMapXCoordinate>2206.0</SVGMapXCoordinate>
      <TimeLastModified>1051890743424</TimeLastModified>
      <GUID>F8F84640FD800549694E2B4C5A6C7198</GUID>
      <EventDate>1051848000000</EventDate>
      <Creator>Tim Ross</Creator>
    </ds:Application>
  </ds:Object>

```

Fig. 2.4. Example of a 7DS report with attributes names, such as “Description”, “Type”, “StartTime”, “TimeLastModified”, and their corresponding values in XML format.

the expectation of successful data access is low. However, estimating this likelihood presents a considerable challenge and the use of advertisements can only provide some hints.

In general, the following parameters impact the power consumption of a network interface:

- size of packets sent and received
- number of packets sent and received
- time the network interface is on

To reduce the power consumption, these parameters need to be kept low.

7DS can employ a simple mechanism that periodically activates the network interface, resulting in an alternation of *on* and *off* intervals, that takes place in an *asynchronous* or *synchronous* manner. In its asynchronous mode, on and off intervals are *equal but not synchronized*, while in synchronous mode, the on and off intervals are *synchronized among hosts*, although *not necessarily equal*.

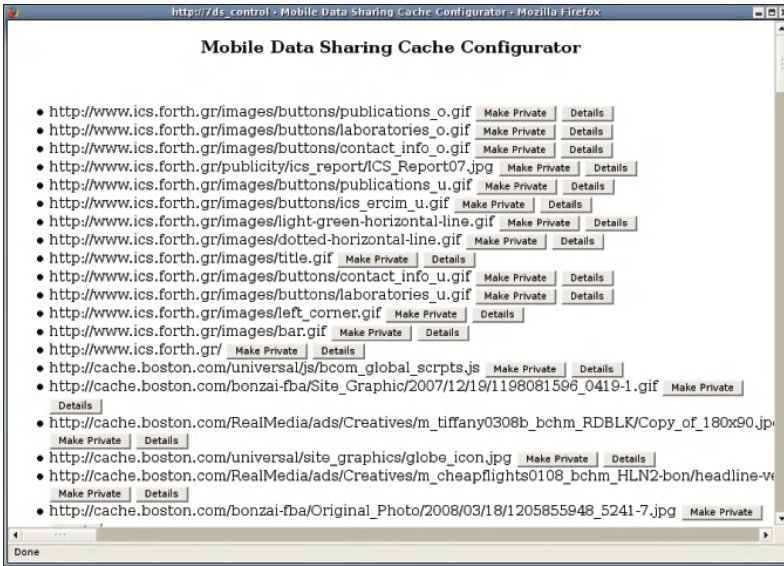


Fig. 2.5. The interface for setting the permission of the cached objects in 7DS.

Hosts may potentially decide on a channel and time interval to communicate and turn their network interface on only during the agreed-upon interval, further reducing the reception of unnecessary traffic. The creation of groups, such that only members of the same group participate in data sharing during the agreed-upon interval and at the specified channel via encrypted messages, may reduce the energy-spending and protect privacy.

Based on the battery level and energy constraints, 7DS may adapt its querying mode (active or passive), type of collaboration (data sharing and forwarding), and power conservation. An evaluation of the different communication patterns is presented in Chapter 3.

When a lower power wireless network interface is available in addition to the IEEE802.11 one, 7DS can use the low-power network interface for the communication between server and peers to decide on data availability, while the IEEE802.11 radio remains mostly off, and is used *only* when a large data object needs to be exchanged.

2.2 Preventing denial-of-service attacks

Mobile devices and wireless networks are vulnerable to different type of attacks aiming to exhaust their resources. One type of such attacks are the denial-of-service attacks that may target different layers. For example:

- Physical layer: creating interference, exhausting the power of devices
- TCP/IP layer: SYN flood, SYN+ACK flood, TCP connection reset attack, bandwidth exhaustion attack
- Overlay network layer: routing attack, eliminating peers by exhausting their power, misbehaving relay devices and caches
- Application layer: application-specific attacks by disseminating false information, storage flooding attack

1. Host Q sends a query
2. Host R receives the query
3. Host R waits for a random time interval T
4. If no challenge for host Q was multicast during T, host R challenges host Q
5. Host Q sends its response
6. Host R verifies host Q's response to the challenge

Fig. 2.6. Responder R challenges querier Q to prevent denial-of-service attacks.

Challenging a host using *hash cash* [69] is a typical method for preventing denial-of-service attacks. These challenges force the host to execute a non-trivial computational task, such as discovering the input in a hash function given the output and a part of the input, before the actual information sharing (Figure 2.6). By challenging the querier at each query, 7DS penalizes malicious users for overloading the network with queries. A potential problem arises when a responder cooperates with a malicious querier, for example, by sending “trivial” challenges or when the querier itself sends “trivial” challenges. To forestall this problem, 7DS can force responders to sign their message, and in that way, other hosts in the wireless LAN can verify the source of the challenge. Furthermore, hosts can use the synchronous approach to reduce the impact of flooding by a malicious user, since hosts will have their network interface on only some specific periods of time—likely unknown to the malicious user.

2.3 Encouraging cooperation

In peer-to-peer systems, cooperation is crucial. In the first generation of peer-to-peer systems, a large percentage of users shared no files [17]. With the exception of BitTorrent, most of the peer-to-peer systems still do not give incentives to users to cooperate. While devices are naturally motivated to cooperate in rescue operations, meetings, or home- or personal-area networks, they may have fewer incentives to collaborate in other environments. This lack of incentives is exacerbated by energy constraints and the possible presence of

selfish or malicious devices that falsely promise to cooperate, disseminate erroneous information, violate the protocols at different layers (e.g., by causing interference and utilizing the shared resources in a selfish manner) or generate denial-of-service attacks [259, 26]. While poor protection of resources can impede the use of a peer-to-peer system, high costs to access the resources can dissuade them. To encourage cooperation, two general approaches have been introduced in literature [37]:

- Micropayment-based (e.g., reward devices for relaying packets or responding to queries)
- Reputation-based (e.g., devices observe behavior of other devices and misbehaving devices are punished)

2.3.1 Micropayment mechanisms

The following micropayment mechanisms can be used in 7DS:

- *electronic checks* (e-checks)
- a *token-based* approach

In e-checks and token-based approaches, nodes remunerate each other for the services they provide to each other. Whereas e-checks do not need trusted hardware, the token-based approach requires a tamper-resistant hardware module in each device for the management of tokens and cryptographic coding of messages, increasing the cost and energy expenditure of mobile devices. Both approaches include an authentication, a micropayment, and an information exchange mechanism.

e-check mechanism

7DS could employ the e-check approach proposed in [88], where hosts sign up for 7DS with a trustee entity or “bank” and acquire an amount of virtual currency as an e-check from that bank. To control the losses from uncollectible transactions, the bank maintains an account limit for each host. As in typical credit models, there is a risk factor, which 7DS can tolerate. e-checks are cryptographically bound to each transaction, which prevents forgery by another host that overhears the exchange of an e-check.

A public-key credential-based architecture can be used: the bank acts as a trusted third party that can authenticate each other offline using appropriate credentials. Each host has its own public key, which is encoded in the credentials along with some restrictions. To minimize losses the credentials are short-lived, and thus, frequently refreshed. 7DS downloads new credentials when the host accesses the Internet, while the bank can limit the amount of micropayments a peer may send to others during a period of time.

The number of credentials issued to a host depends on its usage pattern, service, and trustworthiness. Furthermore, a bank may decline to issue new e-checks or extend the credit line to non-trustworthy hosts. The tradeoff between

reducing the loss and avoiding disruption of cooperation is an interesting research topic.

1. Host R sends its credentials
2. Host Q verifies that host R is known to the bank and is authorized for 7DS
3. Host Q sends an e-check
4. Host Q waits for some time for the data from host R
5. If the time expires, host Q sends a NACK to host R
6. Host R verifies that the e-check is genuine, and
if genuine, host R stores it and sends the data to host Q
7. If host R receives a NACK from host Q, it resends the data to host Q

Fig. 2.7. e-check payment for responding to a query: verification of credentials and e-check exchange.

Let us now assume that 7DS multicast queries are free, but hosts pay to receive the complete data objects after selecting a report that includes a URL to that object. Moreover, let us consider that host Q has multicast a query, and host R has responded by sending a report with the relevant data in its local cache. In its report, host R also indicates the amount of payment required for the transmission of the complete data. Hosts Q and R authenticate each other, and then verify each other's capabilities. Host Q verifies that host R is known to the bank and is authorized to charge host Q's account for this transaction. Host R verifies that host Q is authorized by the bank to proceed with the specific transaction. When a transaction is completed, host Q receives the data object, and host R receives an e-check from host Q. The e-check is encoded as credentials that authorize payment for that specific transaction. Host Q creates its credentials signed with its RSA key [44, 328] and sends them to host R (Figure 2.7).

The credentials include the time they were issued, thereby constraining the amount of payment per responder during a time interval and limiting the risk of double-depositing the e-checks by a responder. Note that there is no guarantee that host R will transmit the data to host Q after receiving host Q's e-check.

The communication between the bank and hosts can take place using established cryptographic protocols, such as IPsec [65, 66]. Periodically, hosts provide their collected e-checks to the bank that, in turn, verifies the transactions and updates the relevant accounts (in the above example, it increases R's account and decreases Q's). The bank can employ the same verification method that host R used to check Q's credentials. Furthermore, it can generate short-term credentials for the host over the secure link, with a new public key being refreshed each time.

An advantage of e-checks is that they do not need trusted hardware. On the other hand, certain constraints discourage cooperation, namely the frequency of contact with the bank in order to upload received e-checks and obtain new ones, the account limit, and expiration of e-checks. The e-check system is designed to tolerate manageable losses, rather than preventing them, and does not provide anonymity.

Token-based micropayment approach

Unlike e-checks, the token-based mechanism assumes the existence of a *secure module* (i.e., trusted and tamper-proof secure hardware), and a *trustee agent* or “bank” that distributes some virtual currency or *tokens*. A token-based micropayment approach was proposed by Buttyan *et al.* [97] to support message relaying in mobile ad-hoc networks. In their system, hosts register with the trustee agent and receive a number of tokens, which are stored in their “purse”, a counter that resides in the secure hardware and indicates the wealth of the host. Tokens come in a single “denomination”, without any monetary value, and can be employed by 7DS to pay hosts that respond to queries.

To prevent a node from illegitimately increasing its own counter, the counter is maintained by the secure module in each node. The tokens that are loaded into the packet are protected from illegitimate modification and detachment from their original packet by cryptographic mechanisms.

A public key infrastructure with public key certificates to verify the public key of a peer can be used. In its secure module, each host keeps its own public and private key, a public key certificate from a certificate authority, and the counter.

1. If counter is not sufficiently loaded, return warning
2. Verify host R's public key certificate, if valid continue
3. Form query
4. Insert query in the list of pending queries
5. Send query to host R
6. If no data sent for pending queries within a predefined time interval,
decrease counter, and send NACK
7. If data received for pending query,
decrease counter, and send ACK

Fig. 2.8. The querier Q runs these steps on its secure module.

As soon as the querier successfully responds to the challenge, the micropayment and data exchange take place. Through an authenticated key agreement protocol, such as the authenticated Diffie-Hellman or Station-to-Station

1. Verify public key certificate. If valid, continue
2. Form response with data
3. Send data
4. If ACK received, increase counter
5. If NACK received, increase counter and resend data

Fig. 2.9. Operations running on the secure module of the responder R.

(STS) protocol [128], the two hosts can establish a shared key. Before sending a query, a host can run the STS protocol, so the parties' key pairs can be generated anew. The public keys are certified, so that the parties can be authenticated. The STS protocol expires after some time, so for each query hosts need to rerun it. An STS channel is established between the secure modules of the two hosts and a shared key is generated to be used for encrypting all messages exchanged between them. Through this secure module, 7DS can prevent hosts from double-spending.

When requesting the complete data object, the querier (host Q) and the responder (host R) perform the operations described in Figures 2.8 and 2.9, respectively, on their secure module.

2.3.2 Reputation mechanisms

In reputation-based trust models, the higher the reputation, the more trustworthy the peer. To avoid malicious peers, peers communicate and share resources with only trustworthy devices. Reputation-based systems require stable identities to hold peers responsible for their actions. However, the creation of multiple identities, the abuse of identities, and the provision of fake or dishonest feedback ratings can be relatively easy in ad-hoc wireless networks. Thus reputation-based systems, and more generally, the provision of security in such networks is arduous due to its offline nature, lack of continuous access to a trustee entity, and power constraints of the devices [370, 195]. Furthermore, resource sharing in 7DS is a relatively short-term exchange. All the above characteristics make the micropayment-based approach more appropriate and simpler to use in the context of 7DS.

2.4 Location-sensing using the peer-to-peer paradigm

Location-sensing has been impelled by the emergence of location-based services in the transportation industry, emergency situations for disaster relief, the entertainment industry, and assistive technology in the medical community. To support location-dependent services, a device needs to estimate its

position. For example, the GPS-enabled navigation systems allow users to compute a route to guide them. However, GPS typically breaks down near obstacles, such as trees and buildings, and does not work indoors.

Location-sensing systems can be classified according to their dependency on and use of:

- specialized infrastructure and hardware
- signal modalities
- training
- methodology and/or use of models for estimating distances, orientation, and position
- coordination system, scale, and location description
- localized or remote computation
- mechanisms for device identification, classification, and recognition
- accuracy and precision requirements

The distance can be estimated using *time of arrival* (e.g., GPS, PinPoint [366]) or *signal-strength* measurements (e.g., Radar [71], Ekahau [13]), if the velocity of the signal and a signal attenuation model for the given environment, respectively, are known. The coordination system can be *absolute* or *relative*, while the location description *physical* or *symbolic*. Accuracy and precision are typical metrics for evaluating a positioning mechanism. A result is considered to be *accurate*, if it is consistent with the *true* or *accepted* value for that result. *Precision* refers to the *repeatability* of a measurement and is an indication of how sharply a result has been defined. It does *not* require us to know the correct or true value. A survey of positioning systems can be found in [183].

Positioning systems may employ different modalities, such as:

- IEEE802.11 (Radar [71, 171], Ubisense [39], Ekahau [13])
- infrared (Active Badge [323])
- ultrasonic (Cricket [301, 302], Active Bat [307])
- Bluetooth [171, 77, 148, 318, 94, 64, 171]
- 4G [322]
- vision (EasyLiving [236, 29])
- physical contact with pressure (Smart Floor), touch sensors or capacitive detectors

A location-sensing system may infer the position using statistical analysis or pattern matching techniques on measurements acquired during a training and run-time phase. The popularity of the IEEE802.11 network, its low deployment cost, and the advantages of using it for both communication and positioning, make it an attractive choice. Most of the signal-strength based localization systems can be classified into the following two categories:

- signature or map-based
- distance-prediction based

The first type creates a *signal-strength signature* or *map* of the physical space during a training phase and compares it with analogous run-time measurements [71, 239, 365]. To build such maps, signal-strength data is gathered from beacons received from APs at various predefined checkpoints during a training phase. Thus, each checkpoint in the map associates the corresponding position of the physical space with statistical measurements based on signal-strength values acquired at those positions. Such maps can be extended with data from different sources or signal modalities, such as ultrasound from deployed sensors to improve location-sensing [301, 171].

In other situations, a dense deployment of a wireless infrastructure for communication and location-sensing may not be feasible due to environmental, cost, and regulatory barriers. Ad hoc networks exploit cooperation by enabling devices to share positioning estimates [327, 185, 104, 275, 116, 146, 366].

CLS is a novel location-sensing system using two features:

- the peer-to-peer paradigm
- probabilistic-based frameworks for transforming measurements from various sources to position and distance estimates

CLS applies the peer-to-peer paradigm by enabling devices to gather positioning information from other neighboring peers, estimate their distance from their peers based on signal-strength measurements, and position themselves accordingly [151]. Periodically, CLS can refine its positioning estimates by incorporating newly received information from other devices.

CLS adopts a *grid-based* representation of the physical space; each cell of the grid corresponds to a physical position in the physical space. The cell size reflects the spatial granularity/scale. Each cell of the grid is associated with a value that indicates the likelihood that the node is in that cell. These values are computed iteratively using one of the following approaches:

- A simple *voting* algorithm, through which a local CLS instance casts *votes* on cells of the grid. A vote on a cell indicates the likelihood that the local device is located in the corresponding area of that cell.
- A particle filter-based model.

CLS can *incorporate additional information* to improve its location estimates. Examples of such information are: position estimates from different network interfaces (e.g., Bluetooth, RF tags, IEEE802.11), contextual semantics (e.g., topological information about the environment, mobility patterns, hotspots of the area), and signal-strength-based signatures of the physical space, to improve the location estimation.

2.4.1 Overview of CLS

CLS aims to enable devices to determine their location in a self-organizing manner without the need for extensive infrastructure or training. The design of CLS was driven by the following desired properties:

- tolerance to multiple network failures (e.g., AP failures or disconnections)
- ability to incorporate application-dependent semantics and various types of measurements
- relatively low computational complexity
- use in both indoor and outdoor environments with pedestrian mobility

CLS can be integrated with a broad range of applications running on devices of different computing capabilities. Some of these devices may have *a priori* knowledge of their location that they can provide reliably. We refer to them as *landmarks*. A device that runs CLS to position itself is referred to as a *node* or *non-landmark* peer.

A node tries to position itself on its local grid through a voting process in which devices participate by sending position information and casting votes on specific cells. Each iteration of a local CLS instance (i.e., running at a peer) consists of the following steps (Algorithm 1):

Algorithm 1 An iteration of the voting process at a CLS instance

1. Gather position information from other peers
 2. Record measurements from the received messages
 3. Transform this information to a probability of being at a certain cell of its local grid
 4. Add this probability to the existing value that this cell already has
 5. Report a position that corresponds to the centroid of the set of cells with maximal weight
-

At the beginning of a run, each peer broadcasts messages to *its one-hop neighbors* that include its *positioning information*, specifically, its local id, maximum wireless range, and position, if known or computed. We refer to this broadcast update as a *positioning message*.

We assume that an AP is configured with its position coordinates and can act as a landmark and send positioning messages in the form of beacons. A peer records the signal strength values with which it receives these messages and responds by broadcasting its own position estimates.

Each local CLS instance transforms these signal-strength values to either distance or position estimates based on a radio attenuation model or a pattern matching algorithm, respectively. Such algorithms relate signal-strength measurements, acquired from messages exchanged between devices, to their position on the terrain or their distance. Based on the position information of the sender and this distance estimation, the receiver estimates its own position on the local grid. When the local CLS estimates its own position, it broadcasts this set of information, i.e., *CLS entry*, to its neighbors. Each node maintains a table with all the received CLS entries. We denote the grid of the node k as G_k and let $v(i, j)$ denote the probability that the cell $(i, j) \in G_k$ is the position of node k . The region of the grid, $G_{h,k}$, i.e., set of cells for which peer k votes as possible region of node h .

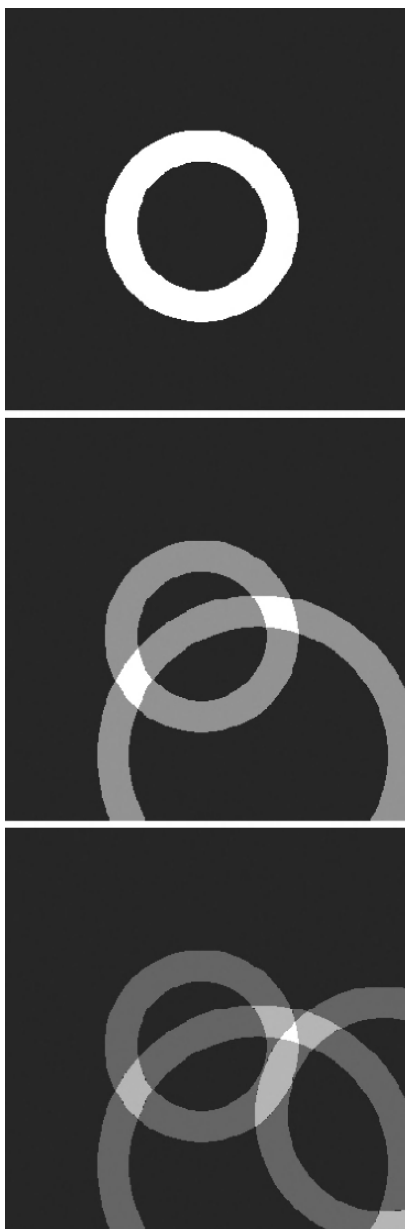


Fig. 2.10. An example of accumulation of votes on grid cells of a host at different time steps. The brighter an area, the more voting weight has been accumulated on the corresponding grid cells. The brightest area corresponds to a potential solution. The grid cell is too small to be distinguishable.

Each node tries to position itself on its local grid. To determine its location, each node h gathers position estimates from other peers, and computes its own location using the Algorithm 2.

Algorithm 2 Position estimation at node h

1. Initialize the values of the grid G_h with all cells containing zeros.
 2. If a signature of the environment is available, compare it with run-time measurements, and for each cell c of the grid, assign a vote of weight $w(c)$ (according to specified criteria).
 3. For each received distance estimation at a peer k with a known or estimated position, perform the following steps:
 - a) Transform the coordinates of peer k to the coordinate system of the grid.
 - b) Determine the region of the grid, $G_{h,k}$, i.e., the set of cells for which peer k votes as possible region of node h . The determination can be based on a position-based or distance-based algorithm. If the peer k is a non-landmark, the distance between the two peers can be computed according to a radio attenuation model or a pattern matching algorithm.
 - c) Increase the value of each cell in $G_{h,k}$ by v_k , where v_k is the voting weight of node k .
 4. Assess the values of the cells in the grid and accept or reject the attempt for location-sensing.
-

This is essentially a voting process, in which a node casts votes on the cells of its grid on behalf of other peers. Votes may have different *weights*. The larger voting weight a cell has acquired, the more likely it is for the corresponding node to be located in that cell. The set of cells in the grid with maximal value indicates the *potential region*. Figure 2.10 shows a snapshot of the grid as three landmarks vote on the location of an unsolved host. The brighter an area, the more voting weight has been accumulated on the corresponding grid cells. The brightest area corresponds to a potential solution.

When a training phase prior to voting is feasible, CLS can build a map or *signature* of a physical space, which is a grid-based structure of the space augmented with measurements from peers. Examples of signal strength-based signatures are: *position-level* and *distance-level* ones. At run-time, a local CLS instance performs the following steps:

1. acquisition of signal-strength measurements from peers
2. creation of a signal-strength map of the space using these measurements
3. generation of a run-time signature
4. comparison of the run-time and training signatures

For the signature comparison, various criteria can be derived based on the statistical characteristics of the signal-strength measurements, such as *confidence intervals* and *percentiles* [345].

Landmarks and nodes that are first to position themselves determine—to some extent—the accuracy of the location estimation of the remaining nodes, since their positioning estimates and errors are propagated in the network through the voting process. To minimize the impact of such errors, CLS imposes the following two conditions:

- The number of votes in each cell of the potential region must be above a threshold. We refer to this threshold as the *solution threshold* (ST).
- The number of cells in the potential region must be below a threshold, denoted as the *local error control threshold* (LECT).

In effect, ST controls how many nodes with known location must agree with the proposed solution. A high ST reduces the error propagation throughout the network, but delays the positioning estimation. On the other hand, LECT determines the precision of each step. Another metric for filtering the local error can be the *diameter* of the region that corresponds to the maximum Euclidean distance of cells with the maximal voting weight.

Additional distance estimates from nodes with known locations increase the voting weight and narrow down the potential region. The values for the ST and LECT could be determined based on network characteristics, such as the *density of nodes* and *landmarks*, and *accuracy of the distance estimations*. To prevent CLS from failing to report a position, both thresholds can be adaptively relaxed after rejecting potential solutions. Once the above conditions are satisfied, CLS reports the centroid of the potential region as the estimated location of the device.

CLS can be implemented in a centralized or distributed fashion, depending on whether or not the computations are performed on a server or peers. Furthermore, in the centralized case one or more servers can be deployed depending on the topography of the terrain.

2.4.2 Particle filter-based framework

In probabilistic terms, CLS can be formulated as the problem of determining the probability of a node being at a certain location given a sequence of signal-strength values. Assuming first-order Markov dynamics, the above problem can be expressed using the network graph depicted in Figure 2.11, where x_k is the node location (system state) at time instant $k = 1, \dots, T$. Notice that x_k cannot be observed directly (it is “hidden”). Besides, for each location x_k , a measurement vector y_k (containing the signal-strength values) is available, that depends on the hidden variable according to a known observation function.

Due to the Markov assumption, each node location, given its immediately previous location, is conditionally independent of all earlier locations, that is

$$P(x_k | x_0, x_1, \dots, x_{k-1}) = P(x_k | x_{k-1}). \quad (2.1)$$

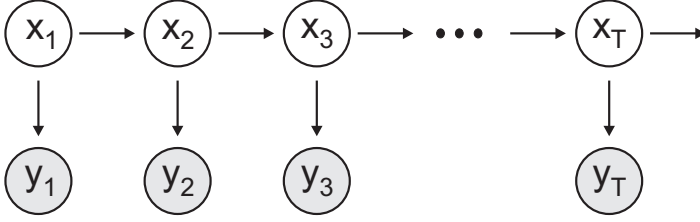


Fig. 2.11. State space model for the proposed location-sensing system. Clear circles indicate hidden state variables, grayed circles indicate observations, horizontal arrows indicate state transition functions and vertical arrows indicate observation functions.

Similarly, the observation at the k -th time instant, given the current state, is conditionally independent of all other states

$$P(y_k | x_0, x_1, \dots, x_k) = P(y_k | x_k). \quad (2.2)$$

Based on this model, location-sensing can be formulated as the problem of computing the location x_k of a node at time k , given the sequence of observations y_1, y_2, \dots, y_k , up to time k , that is, determining the *a posteriori* distribution $P(x_k | y_1, y_2, \dots, y_k)$.

To estimate the above *a posteriori* probability, which is actually a density over the whole state space, we use particle filter. Particle filtering is a technique for implementing a recursive Bayesian filter by Monte Carlo sampling. According to this technique, the *a posteriori* $P(x_k | y_1, y_2, \dots, y_k)$ is expressed as a set of samples

$$x^{(L)} = (x, y)^{(L)}, \quad L = 1, 2, \dots, N \quad (2.3)$$

distributed among the whole state-space. The denser the samples in a certain region of the state-space, the higher the probability that the node is located in that region.

Unlike Kalman filters, particle filters do not impose any constraints on the format of the involved distributions and noise models, or on the linearity of the involved functions. This makes them particularly well-suited to location-sensing.

2.4.3 Performance of CLS and other related systems

Several CLS variants have been implemented and evaluated via extensive simulations and empirical measurements. For the empirical evaluation, we run experiments using IEEE802.11 signal-strength measurements. CLS has a satisfactory accuracy level without the need of specialized hardware and extensive training. It can be easily extended for outdoor environments and different mobility patterns.

We found that the density of landmarks and peers has a dominant impact on positioning. CLS can utilize signal-strength maps of the physical space by superimposing statistical properties of the signal-strength values acquired during the training phase on their corresponding positions. Such maps can significantly improve its performance. Through empirical experiments, we showed how the different statistical properties of signal-strength measurements, the particle-filters model, the AP failures and additional peers affect the performance of CLS. Pre-processing the signal-strength measurements by removing the outliers can further improve the accuracy of CLS.

Currently the training is static, in that it does not consider the placement of rogue or new APs and changes in the configuration, position or orientation of APs and density of users or objects in the area. Such changes may affect the signal-strength values and the signal-strength matching process. A desirable feature is a dynamic calibration phase in which CLS can detect changes in the infrastructure (e.g., position of APs) and incorporate them into the map. The tradeoff between the increased complexity and overhead of the training and runtime phases and the improvements in the accuracy and precision needs to be addressed.

Our simulation results indicate that topological information about the environment (e.g., about hotspot areas, presence information of users, existence of walls, user mobility patterns) can enhance the performance of the system. Part of our future research effort is to incorporate such heuristics into the probabilistic framework of CLS and extend the performance analysis study.

Recently significant work has been published in the area of location-sensing using RF signals. Like CLS, Radar [71] employs signal-strength maps that integrate signal-strength measurements acquired during the training phase from APs at different positions with the physical coordinates of each position. Each measured signal-strength vector is compared against the reference map and the coordinates of the best match will be reported as the estimated position. Bahl *et al.* [72] improved Radar to alleviate side effects that are inherent properties of the signal-strength nature, such as aliasing and multipath.

Ladd *et al.* [239] proposed another location-sensing algorithm that utilizes the IEEE802.11 infrastructure. In its first step, a host employs a probabilistic model to compute the conditional probability of its location for a number of different locations, based on the received signal-strength measurements from nine APs. The second step exploits the limited maximum speed of mobile users to refine the results and reject solutions with a significant change in the location of the mobile host.

Niculescu and Badri Nath [276] designed and evaluated a cooperative location-sensing system that uses specialized hardware for calculating the angle between two hosts in an ad-hoc network. This can be done through antenna arrays or ultrasound receivers. Hosts gather data, estimate their position, and propagate them throughout the network. Previously, these authors [275] introduced a cooperative location-sensing system in which position information of landmarks is propagated towards hosts that are further away, while at the

same time, closer hosts enrich this information by determining their own location. Another location-sensing system in ad-hoc networks performs positioning without the use of landmarks or GPS and presents the tradeoffs among internal parameters of the system [104]. The location-sensing systems presented in [327] and [176] are the closest to CLS and are compared in detail in [151].

Active Badge [351] uses diffuse infrared technology and requires each person to wear a small infrared badge that emits a globally unique identifier every ten seconds or on demand. A central server collects this data from fixed infrared sensors around the building, aggregates it and provides an application programming interface for using the data. The system suffers in the case of fluorescent lighting and direct sunlight, because of the spurious infrared emissions these light sources generate. A different approach, SmartFloor [20], employs a pressure sensor grid installed in all floors to determine presence information. It can determine positions in a building without requiring users to wear tags or carry devices. However, it is not able to specifically identify individuals.

Examples of localization systems that combine multiple technologies are UbiSense [39] and Active Bats [8]. UbiSense can provide a high accuracy using a network of ultra wide band (UWB) sensors installed and connected into a building's existing network. The UWB sensors use Ethernet for timing and synchronization. They detect and react to the position of tags based on time difference of arrival and angle of arrival. An RFTag is a silicon chip that emits an electronic signal in the presence of the energy field created by a *reader* device in proximity. Location can be deduced by considering the last reader to see the card. RFID proximity cards are in widespread use, especially in access control systems. The Active Bats architecture consists of a controller that sends a radio signal and a synchronized reset signal simultaneously to the ceiling sensors using a wired serial network. Bats respond to the radio request with an ultrasonic beacon. Ceiling sensors measure time-of-flight from reset to ultrasonic pulse. Active Bat applies statistical pruning to eliminate erroneous sensor measurements caused by a sensor hearing a reflected pulse instead of one that travelled along the direct path from the Bat to the sensor. A relatively dense deployment of ultrasound sensors in the ceiling can provide within 9 cm of the true position for 95% of the measurements.

Mathematical models that have been used extensively in localization are Kalman, particle filters (e.g., [329, 141, 142, 184]), and Monte Carlo algorithms—also based on particle filters—(e.g., [212, 339, 191, 188]).

2.5 Applications using information sharing via 7DS

To demonstrate the information discovery and caching mechanisms of 7DS, we implemented a prototype and experimented with web browsing and some location-based and collaborative applications.

Customize Wireless Data Sharing Network	
<input type="button" value="Apply"/>	
Update View Time Period (seconds):	<input type="text" value="40"/>
Broadcast Query Time Period (seconds):	<input type="text" value="15"/>
Broadcast Query Limit (broadcasts):	<input type="text" value="10"/>
Retrieve Pages Time Period (seconds):	<input type="text" value="30"/>
Clean cache Time Period (minutes):	<input type="text" value="2"/>
Energy Management ON? (true/false):	<input type="text" value="false"/>
Query Timeout(ADHOC FIRST) (milliseconds):	<input type="text" value="2000"/>
Lucene Indexing (milliseconds) :	<input type="text" value="10000"/>
<input type="button" value="Done"/>	

Fig. 2.12. 7DS configuration. Users can change 7DS parameters via this interface. For example, they can set the frequency that a query is broadcast (BroadcastQuery-Interval) to 15 s, or for web pages without any specified expiration field, the user can set a default one.

The 7DS prototype was written in Java on Linux and also ported to Windows. The Glimpse search engine was used initially but was replaced with Lucene [256], when it became a performance bottleneck. Lucene provides incremental indexing, persistent and non-persistent operations, a built-in lexical analyzer, and a small heap. 7DS peers operate in the ad-hoc mode of IEEE802.11.

Figure 2.12 presents the GUI of the current 7DS implementation that allows users to configure parameters, such as the update view time period, broadcast query time period, and query timeout. The central 7DS interface that displays the queries and corresponding responses is shown in Figure 2.13.

Several aspects of the current 7DS implementation can be improved; For instance, the code is large and complex. However, it can be simplified significantly using libraries included in recent Java versions. All methods required by the applications could be collected to a single class. The time to load web pages—or files for the supported applications—can be also improved. In addition, further experimentation and extension of 7DS to run on smaller devices, such as smart phones and tablet computers, is required.

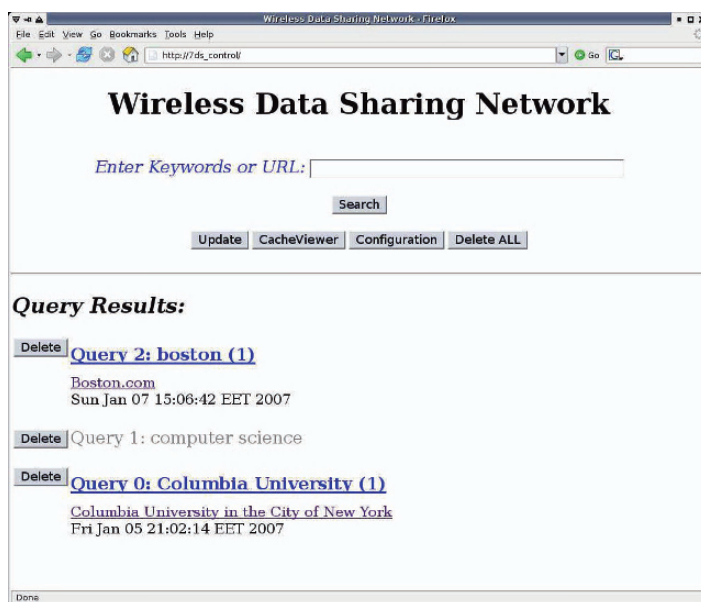


Fig. 2.13. 7DS main interface. In the upper part of the interface, the user can enter a URL or form a keyword-based query or view the cache manager or configuration. Query results are in the lower part. In this example, query 1 is pending, whereas there are responses for queries 0 and 2. Queries 0 and 2 have been expanded, showing their received reports that include URLs.

2.5.1 Web browsing

Although the web is not primarily a location-dependent or collaborative application, it was selected because of its prevalence.

7DS instances share web pages by sending queries containing URLs. After receiving a query, a peer searches its cache, and if a match exists, it forms and broadcasts a report. Such reports can be viewed by the user, who may select the most relevant report and initiate an HTTP GET request to acquire the complete web page (Figure 2.13). Each 7DS instance runs a *miniature web server*, which responds to the HTTP GET requests.

2.5.2 Notesharing and whiteboard tool

The *notesharing* and *whiteboard* applications attempt to improve the collaboration of participants in a seminar, classroom, or meeting by enabling them to circulate a presentation, share and merge their notes for any slide, send queries, and respond to queries. Apart from the core notesharing feature, the

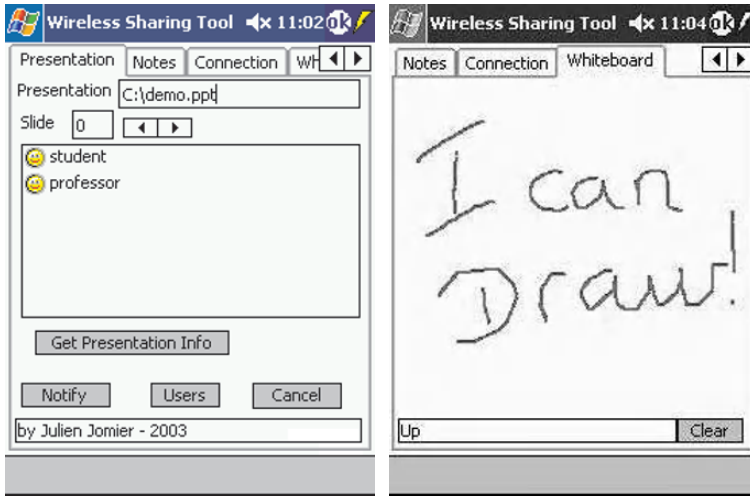


Fig. 2.14. The main interface of the buddylist and whiteboard.

application includes a remote control presentation functionality, buddy list, and virtual whiteboard. Notesharing uses Microsoft PowerPoint—a popular format for presentations—and allows users to take notes for a particular slide.

Let us assume a setting in a classroom with audience and speaker running the notesharing application on their 7DS-enabled devices. By default, the speaker's device is the *master host*, and whenever a slide changes, the system notifies the peers in the (notesharing) multicast group about these changes. Peers may remain synchronized with the current presentation or discard these notification messages. Furthermore, a user may change a slide of the current presentation being displayed by clicking a button on the main user interface. The master host can disseminate the local presentation to peers. Users can search for a specific slide while the application alerts the speaker by changing the color of the title in slides with pending queries.

The buddy list implementation is similar to the messenger-type of applications available on the Internet. When a device joins the multicast group, the name of its local user is added to the list, while it is removed when it leaves, and highlighted when the local user sends a question.

Users can exchange notes in real-time during a meeting. The application maintains an internal list of notes for every slide; objects that include a list of topics, author, slide number, and a description. Once a topic and a description are set for a specific slide, the user may click "submit" to add the notes to the internal list. The new notes are then sent to the multicast group and/or added to the 7DS cache. Notes can be imported from Microsoft PowerPoint, and exported back at the end of the presentation. Users can not only take

notes but also draw pictures on the virtual whiteboard, show their drawing on the current screen, and clear the whiteboard (Figure 2.14).

The notesharing and whiteboard tool was developed using Microsoft Visual C++ on a standard PC and the embedded version with Microsoft Embedded Visual C++. Both versions—desktop and embedded—are based on the Microsoft Foundation Classes (MFC). More information about the notesharing and whiteboard application can be found at [209].

2.5.3 Multimedia traveling journal



Fig. 2.15. The system can superimpose pictures or other multimedia files on a Google map at certain positions that correspond to the locations at which the attached pictures were taken. A marker indicates the number of files associated with that location. A user may click on a photograph to expand it or enter notes.

The *multimedia traveling journal* application enables users to build interactive multimedia journals that associate multimedia files with locations on maps. It runs on top of 7DS, and through 7DS, it allows local peers to share files associated with certain locations. The multimedia files and maps are stored in the cache of the local 7DS instance. A user can add pictures to a certain point on the map by clicking on the map and browsing the image files corresponding to that location (Figure 2.15). Moreover, the user can add, modify, or delete comments on a certain multimedia file, change its permission, and rate its content. A multimedia file can be public or private, and only public files are shared with other peers.

The multimedia traveling journal searches other 7DS peers for multimedia files associated with a given area which has been marked on the map by the user. It forms a 7DS query and multicasts it in the 7DS manner. Furthermore, it maintains and displays the list of neighboring 7DS peers, updating it upon the receipt of a 7DS response. Areas on the map associated with multimedia files can be distinguished by a marker that also indicates the number of the available relevant files.

A user may search for multimedia content related to a certain location in the following manner: First, the user indicates the region of interest by marking the corresponding area on the displayed map (e.g., the white rectangular on the map illustrated in (Figure 2.16). Then, the local 7DS instance will search for relevant data in its cache, on the web, and in the cache of other peers. Specifically, the local 7DS instance will first check its local cache for multimedia files associated with that area. If the search is successful, it will display a marker with a number indicating the number of multimedia files associated with that location. In the case that no relevant data can be found, 7DS's web client attempts to acquire it from the Internet by accessing a predefined web site. Finally, if the web client fails to acquire the requested data (e.g., in the case of intermittent connectivity to the Internet or unavailability of a web server), 7DS will form a *media query* and multicast it to its peers.

The queries are formed using location-based or rate-related criteria. The response of a peer includes the multimedia files, reviews, and ratings and can be displayed. The user frontend of the application is a web browser-based interface that communicates with the local application server using HTTP. It consists of a **Google Maps** map frame on the right and a photo bar on the left side of the window. It employs **JavaScript** and **AJAX**[3, 2] to produce a dynamic and interactive application, instead of just a static web page. Its backend runs on 7DS. It receives all queries from the frontend through 7DS's proxy server, and supports the typical 7DS functionality by adding or deleting photos, querying photos from 7DS neighbors or handing out photos from the local cache. 7DS can also cache **Google Maps** files, enabling the application to work without an Internet connection.

CLS and/or GPS—running as underlying location-sensing mechanisms—periodically record the coordinates of the current position of the device with a timestamp in a positioning trace. Users can upload pictures and videos with their associated timestamp. The multimedia traveling journal can correlate the timestamp information of the multimedia content with the positioning trace and associate the multimedia files with certain areas of a map. The application can also display user's position- or movement-related information on a map, provide “post-it” related functionality, and support various type of devices. Specifically for thin clients (e.g., smart-phones), we implemented the multimedia traveling journal using a more centralized approach, in which a client acquires multimedia files from a predefined web server. A comparative performance analysis on delay characteristics of the peer-to-peer and centralized approaches can be found in [231].

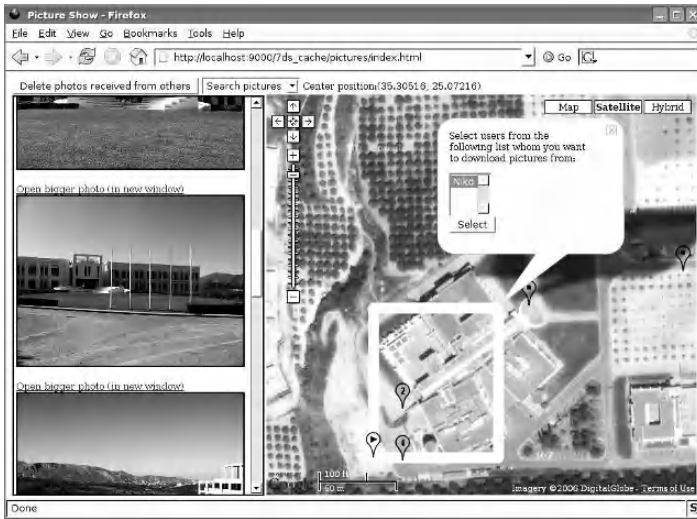


Fig. 2.16. The user can mark the area for which more pictures are requested. Via the local 7DS, the application searches for pictures in the defined area and may select them from a specific user. The local 7DS peers appear in a window.

2.6 Related mobile peer-to-peer computing systems

In the wired WAN domain, several peer-to-peer systems gained popularity in the early 2000s. A non-exhaustive list of them includes: Napster, Gnutella, Freenet, Kazaa, BitTorrent, eDonkey2000, emule, DC++, Morphueus, Bearshare, iMesh, Grokster, Ares, Soulseek, GreenTea, Shwup, and Avalanche [316, 262]. Skype is a voice over IP system that applies the peer-to-peer paradigm with a growing number of users. In these systems, peers are typically stationary clients. Measurement studies have also shown the use of these applications over wireless LANs.

Unlike these systems that are used mostly over wired networks by stationary peers, the United Villages project applies the mobile peer-to-peer paradigm by enabling APs, that are installed on vehicles to transfer data when they are within range of a real-time wireless Internet connection. In the context of relaying, a mesh networking-related company PacketHop recently released a set of specialized software that can be embedded into mobile devices, allowing the device to act as a relay node.

Imielinski and Badrinath were among the first to propose an infrastructure for supplying information services, such as e-mail, fax, and web access to mobile users by placing infostations at traffic lights and airport entrances. Infostations—first mentioned in the context of the DataMan project [303]—

use a single server/multiple clients model in which the server broadcasts data items based on received queries.

As in the case of 7DS, Portolano [138] also aims to provide service discovery to mobile clients with intermittent connections, assuming a hybrid world of a wired infrastructure and wireless links. Its emphasis is on user interfaces that allow mobile clients to discover the semantics of any service and present an interface suited to the client's needs and resource limitations. Odyssey [277] was one of the first platforms designed to enable applications on mobile devices to adapt their media quality based on bandwidth availability without explicit knowledge of one another. For example, when the bandwidth available to a video player drops, it could switch to a video stream with fewer colors and coarser resolution rather than stop completely its transmission. Mobile Chedar [232]—an extension to the Chedar [67] peer-to-peer middleware—provides mechanisms for data streaming between Mobile Chedar nodes and between the Mobile Chedar and Chedar networks. Proem [230] is middleware for developing applications for mobile ad-hoc networks, providing mechanisms for presence and discovery services.

MOBY [187] enables access to services in wide-area networks using the Jini technology. Unlike 7DS—which does not require any registration to an external server—MOBY is based on super-peer architecture: the network is divided into domains by Mnode super-peers. As in a fixed overlay network, the links between Mnodes are preconfigured. LightPeers [119] is a lightweight platform for mobile peer-to-peer networking, targeted to enable mobile devices with limited capabilities to produce, organize, present, and share digital material.

2.7 Conclusions

Peer-to-peer computing manifests several attractive characteristics:

- self-organization, autonomy, and decentralization
- relatively low cost of ownership and sharing by using existing infrastructures and by distributing the maintenance costs
- relatively low cost of accessing resources by enabling resource sharing and low-cost interoperability

To be effective, mobile peer-to-peer computing applications depend on a substantial deployment, cooperation, interoperability, and scalability. In resource allocation, often a tension between cooperation and competition exists. Typically, the scarcer the resources are, the less collaborative the systems tend to be. In other cases, a system may adjust its cooperation-competition policies dynamically depending on the availability of a resource. Given the energy constraints and the nondeterministic characteristics of the environment, such resource allocation algorithms are non-trivial.

While poor protection of resources can impede the use of a peer-to-peer system, high cost and strict conditions to access the resources can dissuade it. The design of a mobile peer-to-peer system needs to balance these two needs. To prevent denial of service attacks, encourage cooperation, and better allocate resources, the use of micropayment- and/or reputation-based mechanisms can be important. However, these mechanisms should have a relatively low overhead, in order to not discourage the active participation of peers. Security and game theory can be applied to address these problems [98].

Increasingly wireless devices collect a large amount of information that can be analyzed to reveal the personal and social context of the user. Such information can be used to support various location-based and context-aware services. At the same time, this abundance of information makes users vulnerable to intrusion of privacy threats. These threats include the identification of the position of the device and potentially, the identity of the subject using the device, which can be acquired directly or inferred using statistical analysis. Malicious users can abuse such information by spamming users with advertisements or disclosing it inappropriately. Thus, a tradeoff between enhancing the information access and disclosing private information inappropriately is exposed. The larger the availability of information, the more likely is to enhance the information access but the larger the vulnerability in privacy threats.

To sustain long-term use, mobile peer-to-peer systems need to be flexible and dynamic and privacy will play an important role in their adoption. Currently, 7DS offers a crude distinction between private and non-private objects and a finer way to describe their privacy requirements is needed. However, privacy is context sensitive and depends on the social context, user activity, ownership of the device, application, and personality of the user. Depending on the context, the system—with or without any user intervention—may decide about the privacy and cooperation policies. Thus, it is critical to provide mechanisms that allow a fine-level description of the privacy requirements and draw a balance between enhancing the service and protecting user privacy.

Information retrieval and querying are at the core of 7DS. Providing semantic-based annotation, discovery, and retrieval of the multimedia information can further enhance the access of information. The development of methods for contextual-knowledge representation and reasoning that involves modeling contextual aspects (e.g., people, devices, locations, and events) is also necessary.

To assist the deployment of mobile peer-to-peer computing systems, a fruitful approach would include the development of the following components:

- a general infrastructure for mobile peer-to-peer applications and a toolkit that new applications could use
- robust mobile peer-to-peer applications with friendly GUIs that can also control the distribution of data and form context- and semantic-based queries
- protocols that ensure anonymity and privacy

- mechanisms that encourage cooperation among peers in an energy-efficient manner

Mobile peer-to-peer computing may enhance the formation of on-line communities of mobile users and create new socio-technological paradigms. The mobile peer-to-peer paradigm—with its distinct feature of cooperation—can be applied to facilitate the information access and sharing among devices for the support of context-aware services. An underlying objective of such services is the recognition and characterization of the users' contexts without interrupting them from their main tasks. This involves research in domains that span from networking and systems to contextual information representation and reasoning, multi-modal user interfaces and graphics. Thus, mobile peer-to-peer computing, combined with context-aware computing, opens up exciting challenges in computer science, demanding interdisciplinary research and innovative paradigms.

Performance analysis of information discovery and dissemination

This chapter focuses on the impact of the wireless coverage range, querying mechanism, density of hosts, and cooperation on information dissemination. It presents performance analysis results acquired via extensive simulations and discusses theoretical models.

3.1 Information discovery schemes

Pervasive computing environments evolve rapidly, encompassing a range of heterogeneous networked systems that have been integrated into physical objects. These environments include a plethora of new human-computer interfaces for seamless interaction across a range of devices, varying from wearable platforms to large displays, sensors, and networked physical objects in interactive rooms, urban settings or rural areas. Examples of wearable platforms include not only laptops and PDAs but also external, on-body sensors, various prosthetics and implantable electronics (e.g., cochlear implants, visual prosthetics, ocular video implants).

Urban environments with users accessing wireless-enabled devices and running location-based services inspired the design of 7DS. We anticipate that such environments—especially during rush hours in a platform of a train or a commercial center or a campus—will manifest high spatial locality of information. More generally, we expect that pervasive computing spaces with wirelessly-enabled physical objects that exhibit high spatial locality of information can apply the peer-to-peer paradigm to enhance their wireless access, particularly in areas with weak signal or limited access to the Internet.

The cooperation among mobile devices ripples throughout in this work. Cooperation in this context is realized through data sharing, querying and data forwarding, relaying messages to an Internet gateway, and caching popular data objects. In general, 7DS devices operate in different modes based on their cooperation strategies, power conservation schemes, and query mechanisms.

7DS-enabled devices can interact either in a *peer-to-peer* (P-P) or *server-to-client* (S-C) manner. In P-P, 7DS-enabled devices cooperate with each other according to the peer-to-peer paradigm. Unlike P-P, S-C is asymmetric: there are 7DS-enabled servers that respond to queries and non-cooperative, potentially resource-constrained clients.

Throughout the text, the term *7DS node* or *7DS host* or simply *host* are used interchangeably to indicate any 7DS-enabled device, and *7DS peer* or simply *peer* any 7DS-enabled device that employs the peer-to-peer paradigm. These different modes of operation allow 7DS to instantiate the different mobile information access schemes when possible, and provide complementary access through peers, when an infrastructure is not available.

A 7DS host acquires the data from the local cache of a peer (P-P) or server (S-C) within its wireless coverage using single-hop multicast. Due to the highly dynamic environment, 7DS does not try to establish permanent caching or service discovery “paths”.

To determine the impact of the various modes of operation on mobile data access, variations on P-P and S-C are also proposed. For example, an extension of S-C, the *hybrid S-C* schemes, allow some types of cooperation among clients. Other P-P schemes enable forwarding of queries or responses to extend their coverage.

7DS employs a simple mechanism that periodically activates the network interface, resulting in an alternation of *on* and *off* intervals that takes place in an asynchronous or synchronous manner (Table 3.1). During the on intervals, nodes may communicate with their peers.

- In asynchronous mode, on and off intervals are *equal but not synchronized*.
- In synchronous mode, the on and off intervals are *synchronized among hosts*, although *not necessarily equal*.

Power conservation	Description of on and off intervals
Asynchronous (default)	equal but not synchronized
Synchronous (“sync”)	not equal but synchronized

Table 3.1. Power conservation schemes in 7DS.

The search for data objects takes place using active or passive querying (as shown in Table 3.2). Clients search for a data object by passively or actively querying for it.

- In *passive querying*, a client multicasts its queries only when it is in the range of an information server. A server announces its presence to clients in its wireless range through *advertisement* messages. Upon the receipt of such an advertisement, a client in passive querying responds by sending its queries.

- In *active querying*, a client *periodically* multicasts its query for a data object until it receives the relevant data. Active querying is the default querying mechanism.

Scheme	Query transmission
Active (default)	Periodic broadcast
Passive	Upon the receipt of an advertisement from a server

Table 3.2. Querying schemes in 7DS.

Depending on the type of cooperation, three variations of P-P are proposed:

- data sharing (DS)
- forwarding (FW)
- both data sharing and forwarding enabled (DS+FW)

When forwarding is enabled, upon the receipt of a query or data, 7DS peers rebroadcast it. To prevent flooding the network, a host ignores the query or data, if it has *already* rebroadcast this query or data during the last ten seconds. For example, host A queries for the data and host B receives host A's query. Assuming that host B does not have the relevant data, it will rebroadcast host A's query. When another host residing in the range of host B (e.g., host C) receives host B's message, it will rebroadcast host A's query, if it does not have any relevant data. Host B will receive the query rebroadcast by host C but will ignore it. In all P-P-based schemes, all nodes are mobile with active querying enabled.

Depending on the mobility of the information server, S-C schemes are classified into two categories:

- mobile information server (MIS)
- fixed information server (FIS)

In straight S-C, clients are mobile, noncooperative, receiving data only from the server via active querying, with the energy conservation mechanism disabled. The hybrid S-C schemes assume passive querying and fixed server. Table 3.3 summarizes the 7DS schemes with their querying mechanism.

Scheme	Cooperation	Server mobility	Options	Querying
FIS	only server	stationary	DS	active
MIS	only server	mobile	DS	active
P-P	all hosts	N/A (no server)	DS, FW, DS+FW	active
Hybrid	server, all clients	stationary	DS, FW, DS+FW	passive

Table 3.3. Summary of the schemes with their querying mechanism.

To investigate the impact of transmission power, cooperation, querying, and energy conservation, P-P and S-C schemes—along with their variants—are evaluated. For example, to examine the impact of the cooperation, we compare P-P with straight S-C. The contrast of MIS with FIS reflects the impact of server mobility, whilst the comparison of DS with DS+FW highlights forwarding. Such performance analysis is amenable to an analytical solution only for very simplified user mobility and interaction patterns. Furthermore, modeling user mobility and interaction is challenging, not only due to the difficulty of setting up large-scale testbeds for empirical studies, but also due to the dependency on the specific environment. Thus, to assess the performance of information dissemination, extensive simulations for different modes of operation of 7DS were performed. In this work, the emphasis is on the short-term behavior of the information dissemination. Its long-term behavior has been studied by another group [254] and a brief summary of their main results is also presented. Preliminary analytical results using diffusion-controlled processes theory are also discussed.

3.2 Simulation assumptions

The simulations are not tied to the 7DS implementation, as we wish to uncover the general trends and prominent parameters. To simplify the analysis, the following assumptions are made:

- There is a single data object to be queried.
- At the beginning of each simulation experiment, *only one* node has the data object, while *all* the remaining ones are interested in acquiring it.
- In S-C, the servers are the original dataholders.

The performance of the data dissemination is evaluated using the following parameters:

- the percentage of hosts that acquire the data as a function of time
- the average delay between sending their first query and successfully receiving the data

Our simulations assume a two-dimensional world, with nodes roaming in a $1\text{ km} \times 1\text{ km}$ area according to the *random waypoint mobility* model. This random walk-based model is frequently used for individual (pedestrian) movement [92, 324, 360]. The random waypoint breaks the movement of a mobile host into alternating motion and rest periods. Each mobile host starts from a different position and moves to a new randomly chosen destination. The coordinates of a destination are selected according to a uniform distribution from the interval $[0, 1)\text{ km}$. Each node moves to its destination with a constant speed selected randomly from a uniform distribution in the interval $(0, 1.5)\text{ m/s}$. When a mobile host reaches its destination, it pauses for a fixed

amount of time, then chooses a new destination and speed (as in the previous step) and continues moving.

The query interval consists of an *on* and *off* interval. The broadcast is scheduled at a random time during the *on* interval. The asynchronous mode is the default power conservation method. In schemes with no power conservation, the *off* interval is equal to 0 and there is a concurrence of the *on* and query intervals during which the exchange of queries, reports, and advertisements takes place. A cooperative dataholder responds to a query by sending the data object.

As all simulations assume one data object, the host density reflects the popularity of the data. By varying this density, the impact of the popularity of the data can be highlighted. For example, a density of ten nodes per square kilometer may correspond to the dissemination of a local news article across users with wireless-enabled devices during a rush hour at Grand Central Station in Manhattan.

We used the ns-2 simulator [144] with the mobility and wireless extensions from the CMU Monarch project [54]. 300 different scenarios were generated, each defining the distribution, movement, wireless range, and type of each host that participates in an experiment. Simulations were run using these scenarios, for the different schemes of Table 3.3.

The radio propagation P_r is based on the two-ray ground reflection model, in which the received power at a distance d is estimated by

$$P_r(d) = \frac{P_t G_t G_r h_r^2 h_t^2}{d^4} \quad (3.1)$$

where P_t is the power of transmitted signal, h_r and h_t are the heights of receiver and transmitter antenna, respectively, and G_r and G_t are the gains of signal at the receiver and transmitter, respectively [311]. Varying the transmission power, through the high, medium, and low levels, the resultant wireless ranges are approximately 230 m, 115 m, and 57.5 m (Table 3.4). The wireless LAN is based on IEEE802.11.

Parameter	Value
Pause time	50 s
Mobile user speed	(0,1.5) m/s
Server advertisement interval	10 s
Forward message interval	10 s
Transmission power	281.8 (high), $281.8/2^4$ (medium), $281.8/2^8$ (low) mW
Wireless ranges (trx power)	230 (high), 115 (medium), 57.5 (low) m

Table 3.4. Simulation parameters in 7DS.

Parameter	Value
Power conservation	Asynchronous (on, off periods equal to 7.5 s)
Query interval	15 s
Simulation time	25 min
Shape of the environment	1 km \times 1 km

Table 3.5. Default setting in 7DS simulations.

3.3 Data dissemination benchmarks

The performance analysis focused on the following benchmarks:

- the percentage of nodes that acquire the data object (i.e., have become dataholders) as a function of time
- the average delay for a mobile host to receive the data objects since the transmission of its first query

The percentage of new dataholders reported in the plots was computed *excluding* the original dataholder. Only these dataholders were considered for computing the average delay.

To explore the temporal evolution of data diffusion, the simulation time was varied from 25 minutes to 50 minutes. The 95% confidence interval for the average percentage of dataholders is within 0-11% of the computed average, with the variance tending to be higher for low host density.

3.4 Density of dataholders

7DS proves to be an effective data dissemination tool for high transmission power. Even for a sparse network, 77% of nodes will acquire the data during the experiment, while for denser networks, this percentage becomes 96% or more. The effect of cooperation can be highlighted by the comparison of P-P with FIS. Figures 3.1 and 3.2 show the percentage of dataholders as a function of the density of hosts in P-P and S-C with a query interval of 15 s. For example, in a setting of 25 hosts, P-P outperforms FIS by 55%. In particular, in P-P, 99.9% of hosts will acquire the data after 25 minutes, compared to 42% of the users in FIS. For lower transmission power, P-P outperforms FIS by up to 70% (e.g., DS for 25 peers and medium transmission power, as shown in Figure 3.2). The impact of data sharing among peers is also apparent in hybrid schemes, becoming more evident in settings with ten hosts or more per square kilometer and medium or high transmission power.

Forwarding in addition to data sharing does not result in any substantial performance improvement due to the low probability for a querier to reach a dataholder during a simulation run *only* via a multi-hop neighbor. A *nth-hop neighbor* of host A is any host B that can reach host A by *at least* n hops,

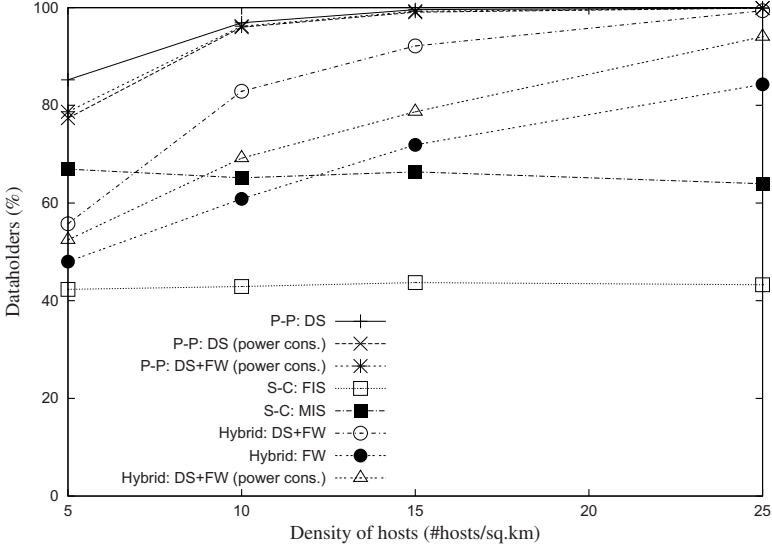


Fig. 3.1. Percentage of dataholders after 25 minutes for high transmission power.

which are relay hosts different from A and B. In settings with a larger number of relay hosts, the impact of forwarding is expected to be more significant. Forwarding without data sharing also improves performance. For example, hybrid schemes with forwarding-enabled outperform FIS by up to 40%, depending on transmission power and peer density.

The performance of P-P improves substantially as the number of hosts increases. On the other hand, the performance of FIS and MIS remains constant as the number of hosts increases, given that a data exchange takes place only when a querier is in close proximity to the server. Depending on the transmission power, MIS outperforms FIS by approximately 22%, 16%, and 6%, respectively. The only difference between MIS and FIS is the mobility of the server, and thus, the relative speed between the server and a client. Due to the higher relative speed, a client is in the range of the server more frequently in MIS than in FIS and thus acquires the data faster. Empirical studies have shown that wireless network interfaces consume substantial power even in an idle state [147]. Asynchronous energy conservation results in a 50% energy savings but also some degradation in data dissemination, as the network interface is on only half the time. Figures 3.1 and 3.2 illustrate the—relatively small—degradation in data dissemination due to the reduced time interval in which hosts can communicate.

High transmission power with the default power conservation mode performs better than medium transmission power without any power conserva-

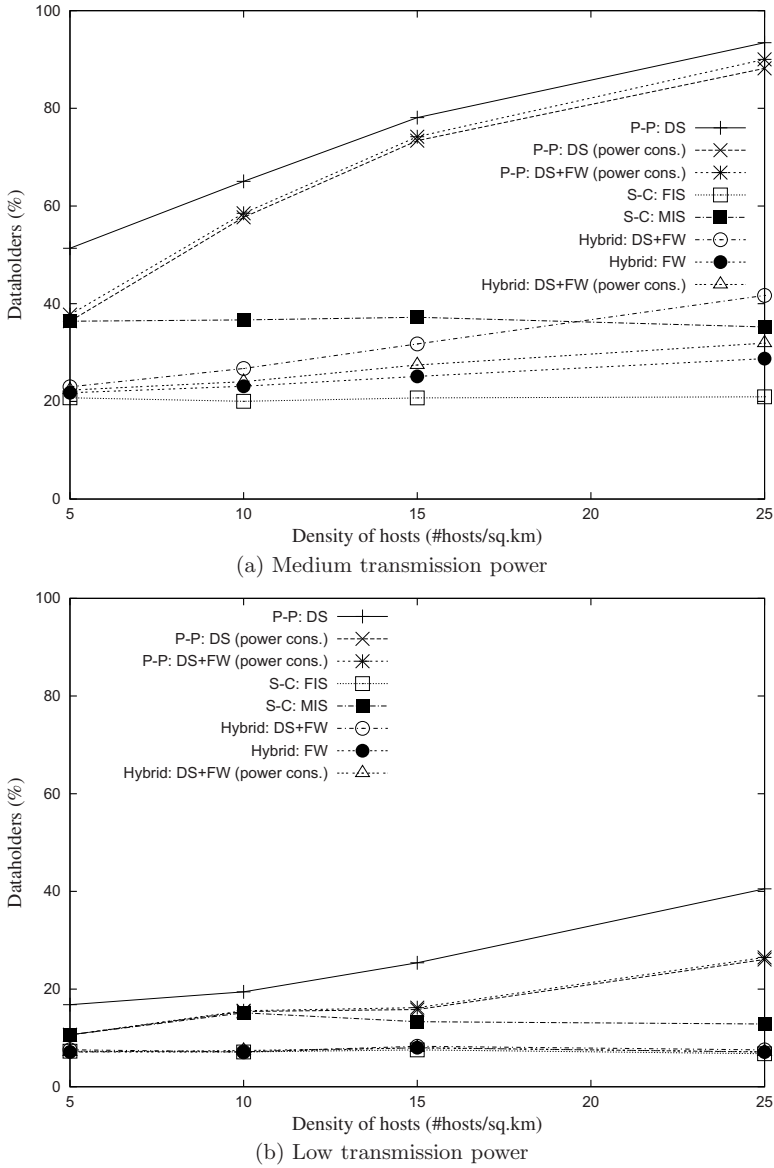
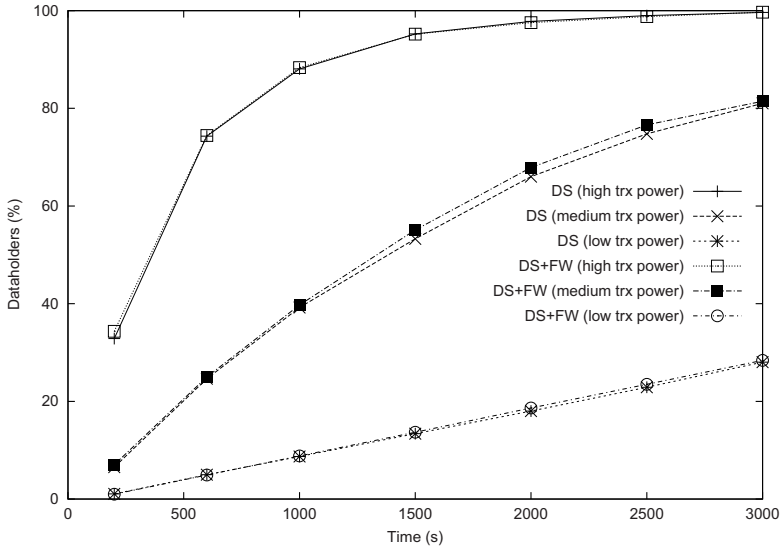
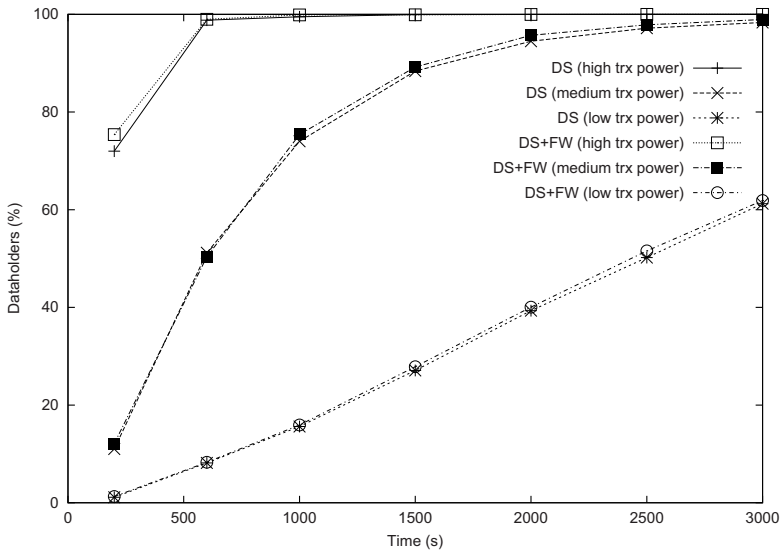


Fig. 3.2. Percentage of dataholders after 25 minutes.



(a) Ten cooperative hosts



(b) Twenty five cooperative hosts

Fig. 3.3. Effect of forwarding on density of dataholders in peer-to-peer with data sharing enabled (DS).

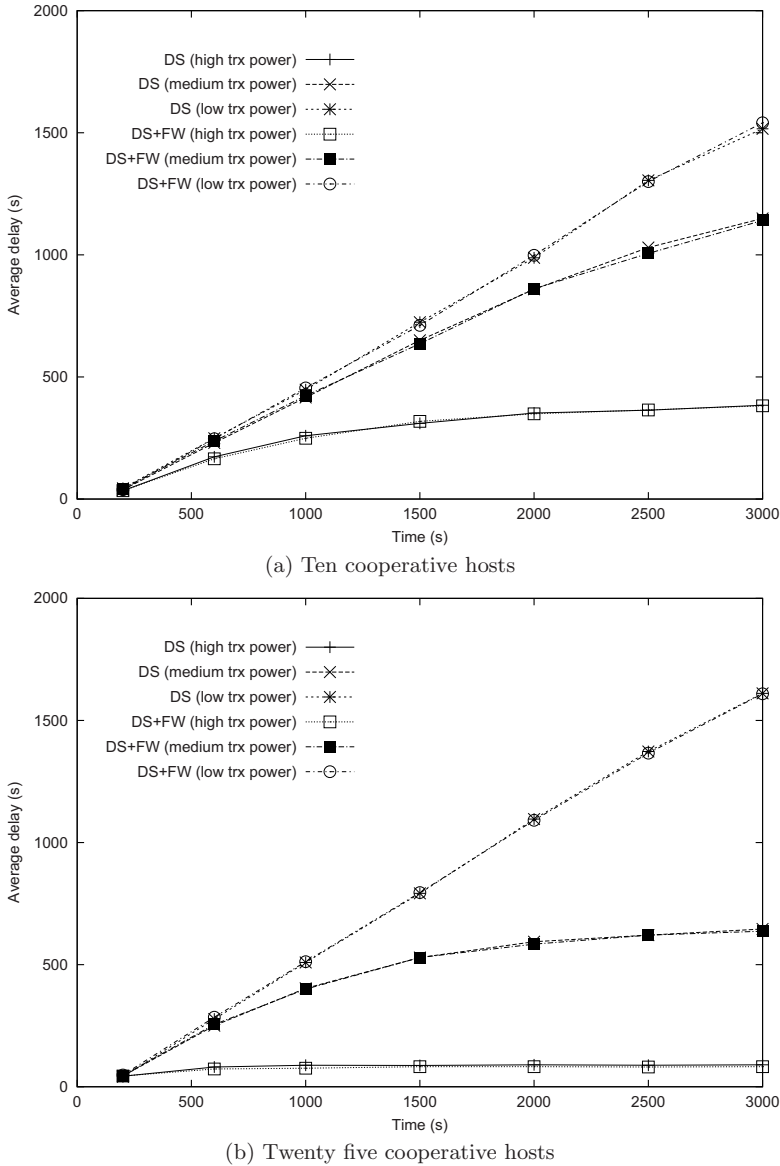


Fig. 3.4. Effect of forwarding on delay in peer-to-peer with data sharing enabled (DS).

tion, e.g., “P-P:DS (power cons.)” in Figure 3.1 compared to “P-P:DS” in Figure 3.2 (a). In general, for a fixed query interval, the smaller the *on* interval, the higher the energy savings but also the larger the degradation of data dissemination. To ameliorate the performance, the synchronous mode can be enabled, where the on and off intervals of all hosts are synchronized. In cases that devices experience congestion or interference, the availability of a lower power network interface for control packets, in addition to the regular one for data exchange, and the efficient channel assignment to groups of peers that intend to cooperate can be crucial.

Let us now highlight the performance of data dissemination as a function of the query interval. Its degradation as the query interval increases is relatively small in FIS compared to P-P due to the fact that opportunities of data exchange occur less frequently in FIS than in P-P: a querier will be in the range of a dataholder less frequently in FIS than in P-P. Figures 3.5 (a) and 3.6 correspond to a relatively sparse network of five hosts per square kilometer while Figures 3.5 (b) and 3.7 show the results for a denser network consisting of 25 hosts per square kilometer. In P-P, the impact of the query interval is more prominent. For example, in the case of medium transmission power, when the query interval increases from 15 seconds to 3 minutes, the degradation in the performance of data dissemination is approximately 17%. Further analysis to estimate the optimal querying mechanism taking into consideration the traffic in the wireless LAN and host co-residency time is required.

3.5 Average delay

An important performance metric is the average delay a host experiences from the first query until it receives the data. For each test, the average delay of the nodes that acquired the data by the end of simulation was computed, considering only the hosts that had received the data by the end of the simulation. The average of all 300 sets—excluding the ones without new dataholders—was reported. For a 25-minute simulation time, the average delay as a function of the probability of acquiring the data was computed.

In P-P with data sharing, no energy conservation, and high transmission power, the average delay is as high as 6 minutes for sparse networks and drops to 77 seconds for dense networks, while for low transmission power, it climbs to 13 minutes. For high transmission power in FIS, it is 6 minutes, while for low transmission power, it reaches 9 minutes.

Figures 3.8, 3.9, and 3.10 compare FIS and P-P with data sharing and no power conservation enabled. To attain these figures, the simulation results for the probability that a host acquires the data, and the average delay it experiences have been combined. For example, in the case of one server in a square kilometer area with high transmission power, each “point” (data entry) of the curves in Figure 3.8 corresponds to a *distinct simulation time*. Each point combines the percentage of dataholders at the corresponding simulation

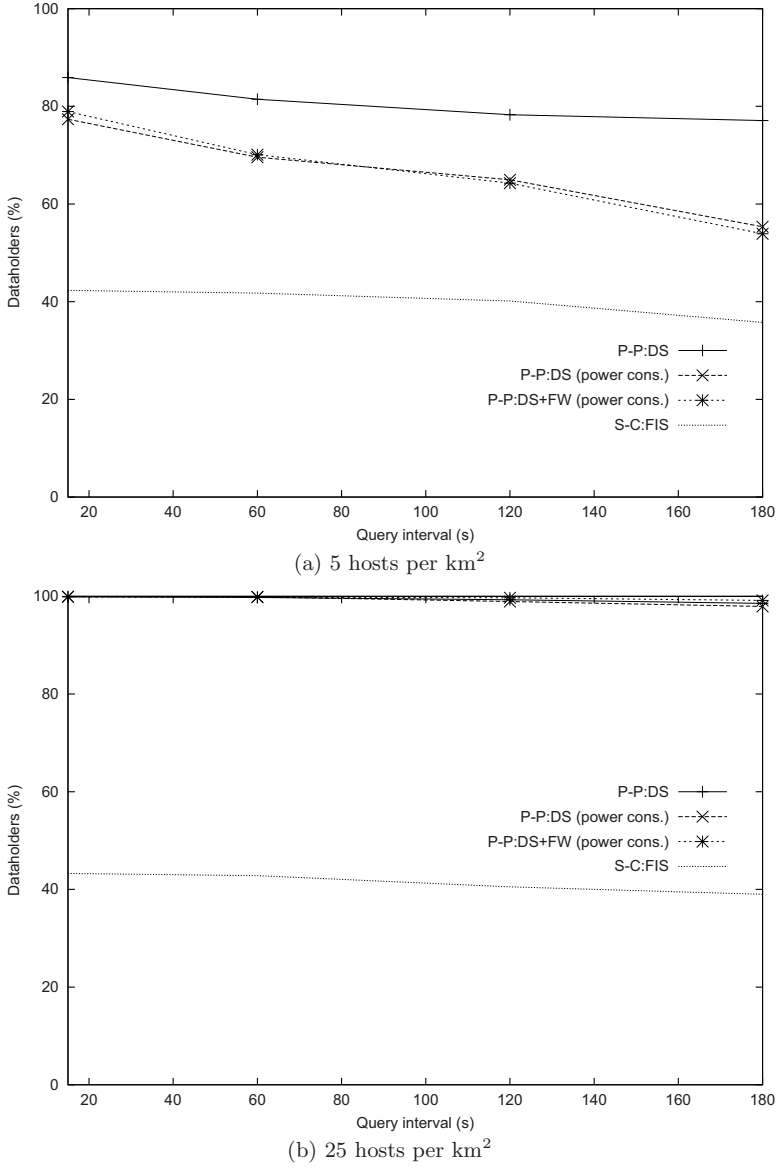
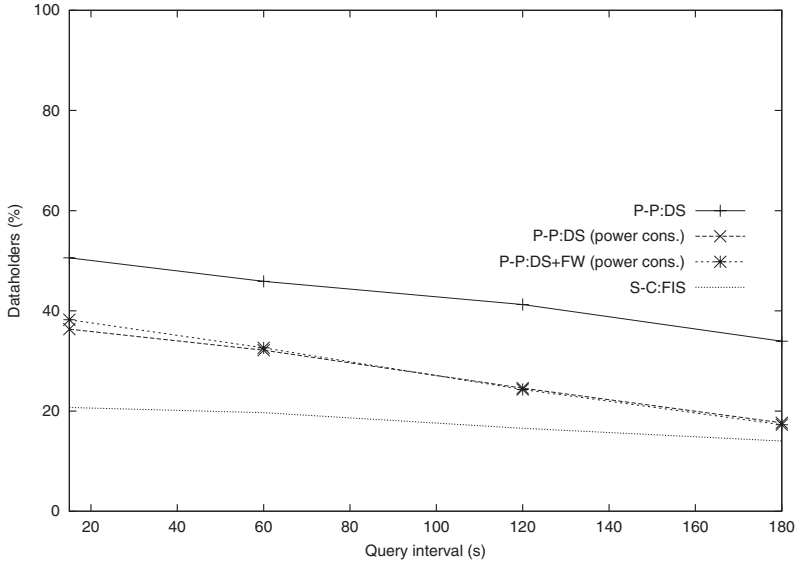
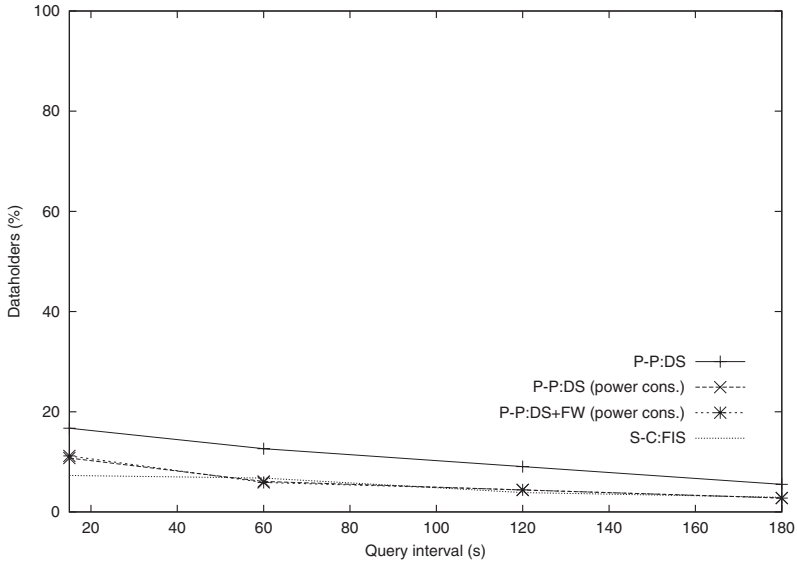


Fig. 3.5. Percentage of dataholders as a function of the query interval. Schemes with power conservation enabled use the asynchronous mode (default). All devices use high transmission power.

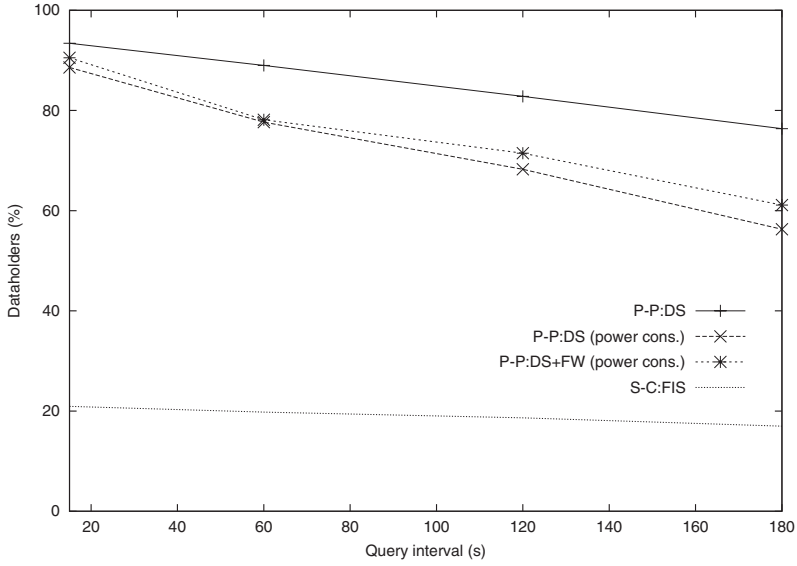


(a) Medium transmission power

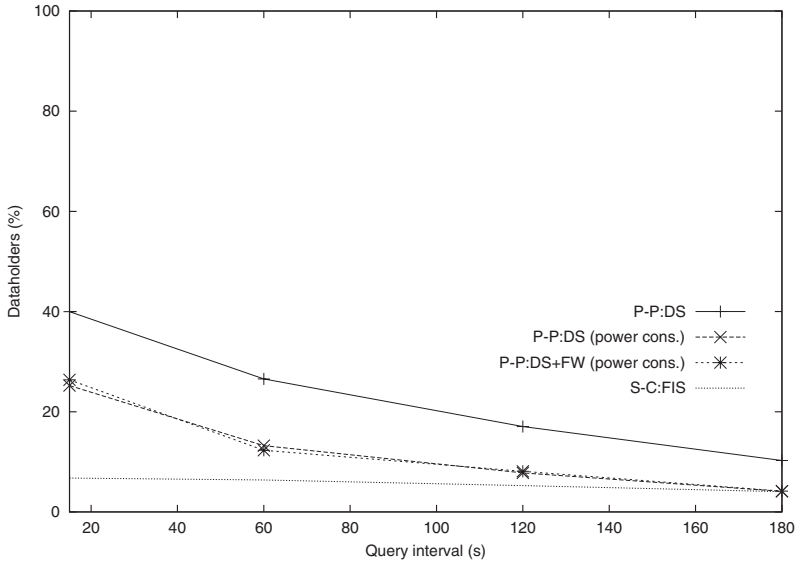


(b) Low transmission power

Fig. 3.6. Percentage of dataholders as a function of the query interval with five hosts per km^2 . Schemes with power conservation enabled use the asynchronous mode (default).

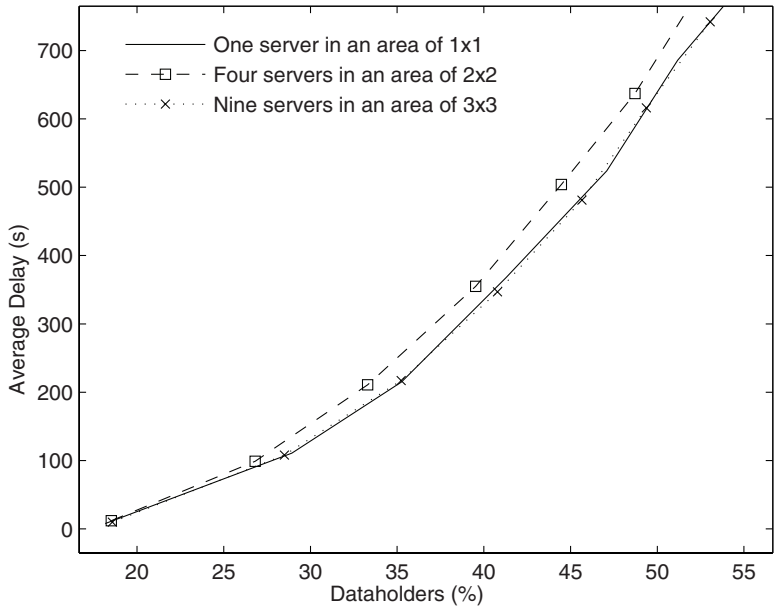


(a) Medium transmission power

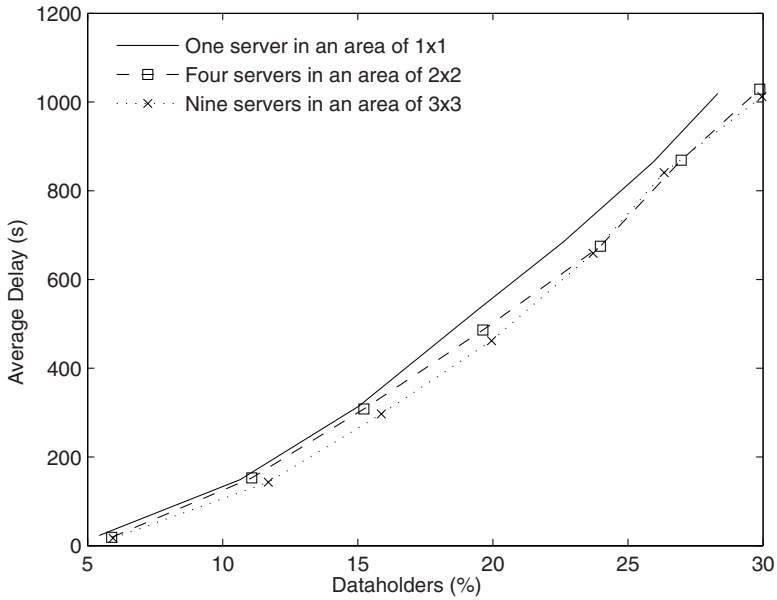


(b) Low transmission power

Fig. 3.7. Percentage of dataholders as a function of the query interval with 25 hosts per km^2 . Schemes with power conservation enabled use the default mode.



(a) All hosts with high transmission power



(b) All hosts with medium transmission power

Fig. 3.8. Scaling property in FIS: fixed density of servers. Average delay of FIS as a function of the percentage of dataholders with one server per square kilometer.

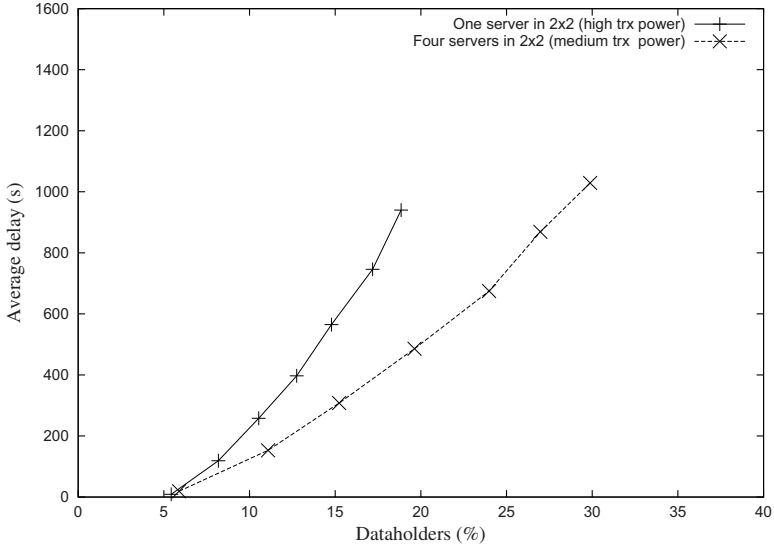
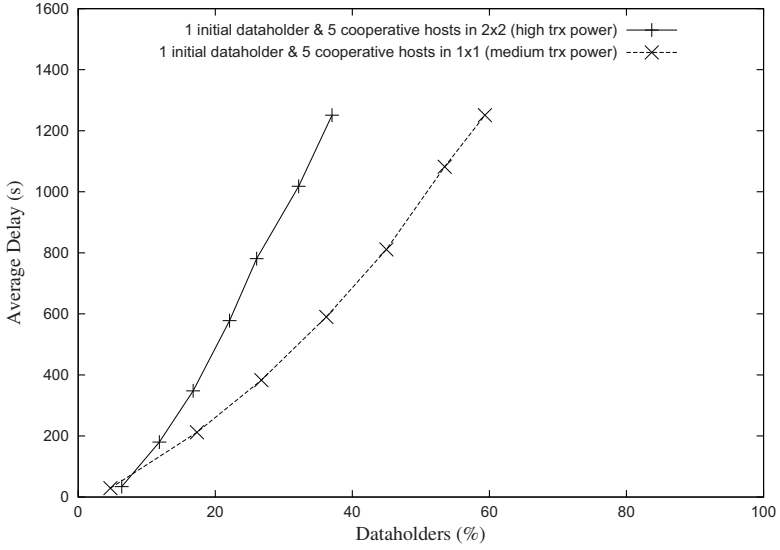


Fig. 3.9. Scaling property in FIS: fixed *total wireless coverage of servers*. Average delay to receive the data and percentage of dataholders for different densities of information servers.

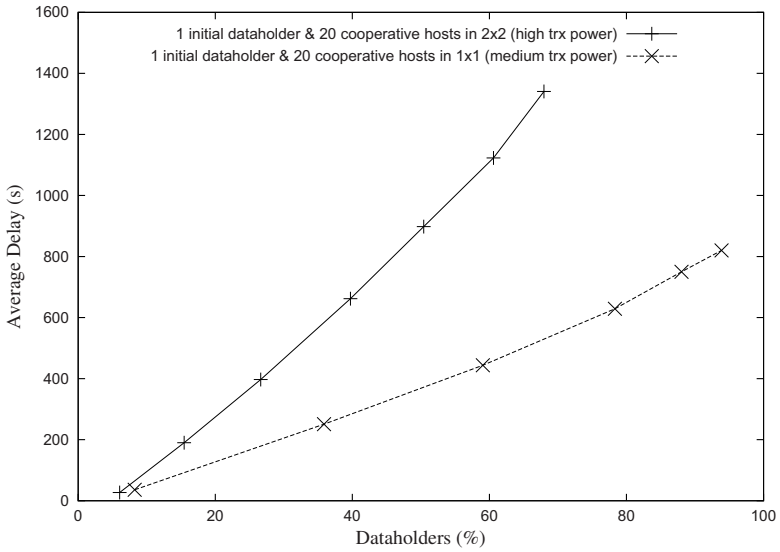
time and their average delay until they become dataholders. The percentage of hosts that acquire the data in P-P with high transmission power reaches 40% with an average delay of 135 s while for the same delay, 30% of hosts will acquire the data in FIS. In FIS, a 40% probability of acquiring data corresponds to an average delay of 6 minutes. For a higher average delay of 10 minutes, 85% of hosts will acquire the data using P-P, and 50% using FIS. In the case of medium transmission power, with an average delay of 315 s, a host will get the data with a probability of 15% and 22% using FIS and P-P, respectively.

3.6 Scaling properties of data dissemination

In both P-P and FIS, when the area is expanded but the density of hosts and their transmission power are kept fixed, the performance of data dissemination remains the same, indicating the robustness of our simulation results. Figure 3.8 shows this scaling property in FIS for high and medium transmission power.



(a) 5 cooperative hosts per square kilometer



(b) 20 cooperative hosts per square kilometer

Fig. 3.10. Average delay to receive the data as a function of the percentage of dataholders in P-P with data sharing schemes.

Another interesting scaling property is related to the effect of the density of cooperative hosts and their wireless coverage density, whilst keeping the total area of wireless coverage fixed.

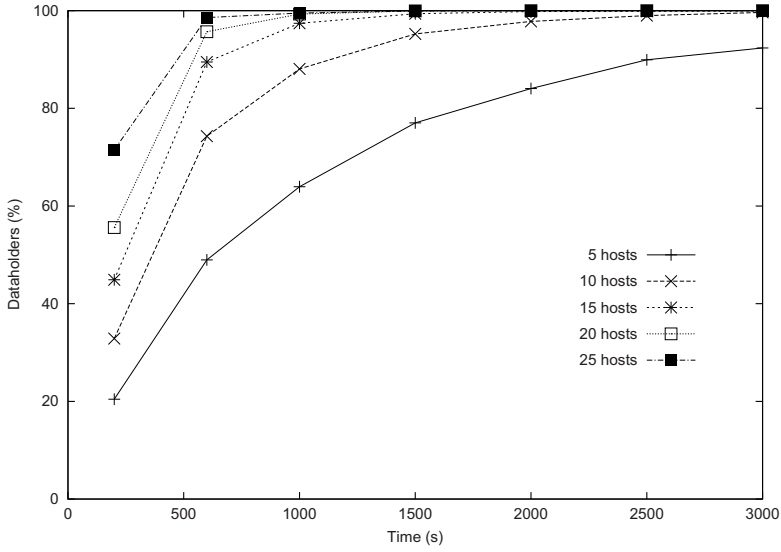
Let us assume two deployments of servers with different density of servers and transmission power but of the same aggregate wireless coverage. For simplicity, let us also consider the free-space model for the radio communication. The deployment of the larger density of servers is more effective in terms of power expense and wireless throughput utilization. We found that for fixed total wireless coverage, the higher the density of cooperative hosts, the better the performance in FIS and P-P. An intuitive explanation is that the two deployments become equivalent by “scaling down” the deployment of the lower density of servers to match the other. After this “scaling”, the speed of the hosts in the deployment with the initially lower density of servers scheme doubles. Thus, this setting “becomes” the same as the other one in terms of area, transmission coverage of each server, and server density but with hosts moving faster. Therefore, the probability of a host to get into the coverage area of a server increases.

Figure 3.9 compares two FIS settings with the same total wireless coverage density of cooperative hosts (servers). The first includes one server in a $2\text{ km} \times 2\text{ km}$ area with high transmission power and the latter four servers in a $2\text{ km} \times 2\text{ km}$ area with medium transmission power. The setting with a higher density of servers performs better. For example, for a 20% probability of acquiring the data, FIS with a higher density of servers produces an average delay of 500s. For the same wireless coverage but with a lower density of servers, the average delay doubles. Similar phenomena holds for P-P, as illustrated in Figure 3.10 for various host densities.

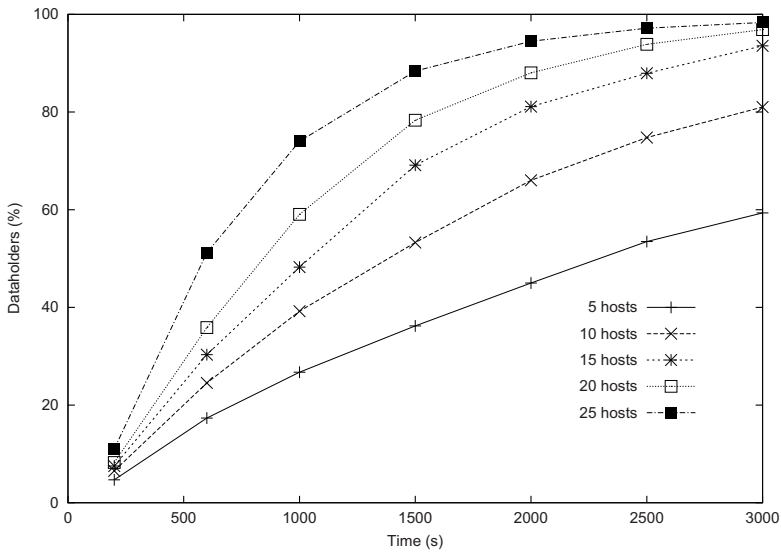
3.7 Models of information dissemination

This section discusses our initial efforts to analytically study the wireless data dissemination and further generalize our simulation results. Information dissemination can be realized through gossiping algorithms, which have been studied analytically. For example, Ravishankar and Singh [312, 314, 313] presented an optimal broadcasting algorithm, considering a one-dimensional world in which nodes are placed on a line. Percolation theory [131] has been also employed for estimating the expected time for a message to spread among all nodes placed on a lattice. Such studies use the shape theorem that typically assumes a system in two-dimensional co-ordinates, in which each lattice site is either empty or occupied, and in which the set of occupied sites A_t at time t grows and attains a limiting geometry. Simple epidemic models have also appeared in [283] and diffusion-controlled processes in [284, 286].

Section 3.7.1 presents a simplistic epidemic model and Section 3.7.2 discusses a novel approach to model data dissemination borrowed from particle kinetics as well as diffusion-controlled processes.



(a) High transmission power



(b) Medium transmission power

Fig. 3.11. Performance of P-P with DS and asynchronous power conservation enabled (default mode) as a function of simulation time and various cooperative host densities.

3.7.1 Simple epidemic model

Mathematical models for the spread of epidemic diseases have been widely studied [73, 140]. In our case, a disease is equivalent to a data object. These models analyze the fraction of infected individuals (i.e., mobile nodes) among a finite population (e.g., an ad-hoc network) and the probability with which the entire population is infected after a given time.

7DS aims to prefetch and disseminate data for mobile hosts not necessarily connected to the Internet. Its effectiveness, as a data dissemination and prefetching tool, depends on a variety of parameters, such as node density in a certain region, node mobility, transmission power, cooperation strategy, querying mode, and energy conservation. It does not appear to be amenable to an analytical solution except for simplified versions.

The assumption that in any time interval h , any given dataholder will transmit data to a querier with probability $h\alpha + o(h)$ can substantially reduce the complexity. Note that in order for a function $f(\cdot)$ to be $o(h)$, it is necessary that the limit of $f(h)/h$ is equal to zero as h goes to zero. But if h goes to zero, the only way for $f(h)/h$ to approach zero is for $f(h)$ to go to zero faster than h does. That is, for h small, $f(h)$ must be small compared with h [319].

A simple epidemic model can be then used to compute the expected delay for a message to be propagated to the population of an area, as described in [321]. For the epidemic model, the following assumptions are made:

- A population of N peers at time 0 consists of one dataholder (the “infected” node) and $N - 1$ queriers (the “susceptibles” ones).
- Once a peer acquires the data, the data will be locally stored permanently.
- In any time interval h , any given dataholder will transmit data to a querier with probability $h\alpha + o(h)$.

If $X(t)$ denotes the number of data holders in the population at time t , the process $\{X(t), 0 \leq t\}$ is a pure birth process with rate λ_k

$$\lambda_k = \begin{cases} (N - k)N\alpha, & k = 1, \dots, N - 1 \\ 0, & \text{otherwise} \end{cases} \quad (3.2)$$

Thus, when there are k dataholders, each of the remaining mobiles will get the data at a rate equal to $k\alpha$. If T denotes the time until the data has been spread amongst all the mobiles, then T can be represented as

$$T = \sum_{i=1}^{N-1} T_i, \quad (3.3)$$

where T_i is the time to go from i to $i + 1$ dataholders. As the T_i are independent exponential random variables with respective rates $\lambda_i = (m - i)\alpha$, $i = 1, \dots, m - 1$, the expected value of T is given by

$$E[T] = \frac{1}{\alpha} \sum_{i=1}^{N-1} \frac{1}{i(N - 1)}. \quad (3.4)$$

3.7.2 Diffusion-controlled process

We apply a theoretical framework based on diffusion-controlled processes, random walks and kinetics of diffusion-controlled chemical processes [280] to model FIS. Let us first describe a diffusion process that is closely related to information dissemination. Consider a diffusion process that takes place in a medium with randomly distributed static traps and two types of particles, namely, S-type (stationary traps or sinks) and M-type (mobile particles) [196]. In such a static trapping model, particles of M-type perform diffusive motion in d -dimensional space while particles of S-type are static and randomly distributed in space. M-type particles are absorbed by S-type when they collide with them. The simple trapping model assumes traps of infinite capacity. The diffusion-controlled processes focus on the *survival probability*, that is the probability that a particle will *not* get trapped as a function of time.

Rosenstock's trapping model in d dimensions assumes a genuinely d -dimensional, unbiased walk of finite mean-square displacement per step and has a survival probability ϕ_t that for large t follows

$$\log(\phi_t) \approx -\alpha \left[\log \left(\frac{1}{1-q} \right) \right]^{(\frac{2}{d+2})} t^{(\frac{d}{d+2})} \quad (3.5)$$

where α is a lattice-dependent constant, and q denotes the concentration of the independently distributed, irreversible traps.

One question is: when Eq. 3.5 is a useful approximation? To answer this question, most studies have relied on simulations, but so far there is no information available on the range of validity of Eq. 3.5. In [174], Havlin *et al.* presented evidence suggesting that Eq. 3.5 is a useful approximation when

$$\rho > 10 \quad (3.6)$$

where ρ is the scaling function

$$\rho = \ln \left(\frac{1}{1-q} \right)^{\frac{2}{d+2}} t^{\frac{d}{d+2}}. \quad (3.7)$$

This value of ρ corresponds to a survival probability which is equal to 10^{-13} in both two- or three-dimensional spaces. Havlin *et al.* argued that pure simulation techniques will always lead to an exponential decay at sufficiently long times, rather than to the correct decay given by the theoretically-proven Eq. 3.5. Their evidence for the new lower value of ρ is based on two numerical techniques that they developed. One of these is practical for high trap concentrations only ($q \geq 0.9$). This case of high trap concentrations is analogous to our case.

Information sharing in FIS takes place between the server and the querier. When a 7DS querier is in the range of the server, it acquires the data. It is easy to draw the analogy between FIS and the trapping model:

- The stationary information servers can be modeled as S-type particles (stationary traps) and the mobile clients as M-type particles (mobile particles).
- A data acquisition “corresponds” to a trapping event. When a querier acquires the data (or “an M-type particle gets trapped”), it remains a dataholder (or “is trapped”) for the remaining time, and thus the survival probability corresponds to the probability of not acquiring the data.
- The term $1 - \phi_t$ expresses the fraction of hosts that acquire the data at time t .

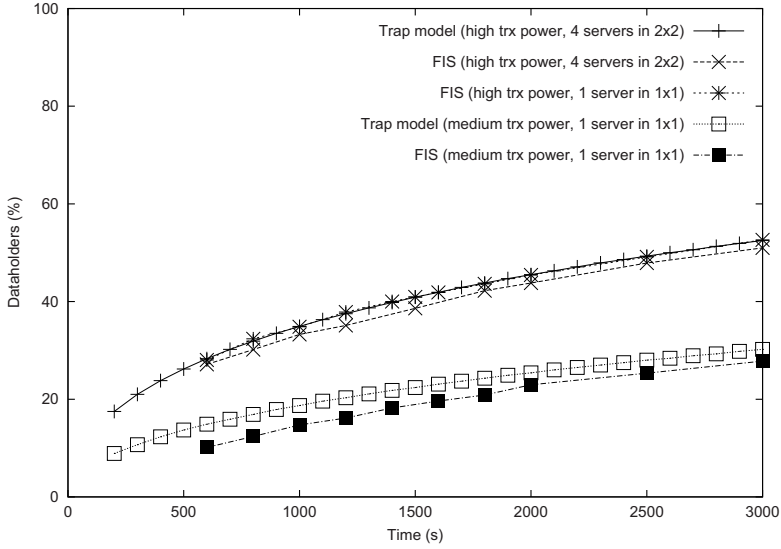


Fig. 3.12. Simulation (FIS) and analytical trapping model results.

Figure 3.12 illustrates the analytical and simulation results for data dissemination as a function of time. The analytical results for the trapping model are derived from Eq. 3.5 (Rosenstock’s trapping model) for high and medium transmission power.

Let us define the wireless density of servers q as $\pi R^2 N_s / A^2$, where N_s is the number of servers placed in an area of size $A \times A$, and R is the wireless range equal to 230 m and 115 m for high and medium wireless range, respectively. For the results in Figure 3.12, we used the FIS simulations, described in Section 3.4.

To investigate if our simulation results were consistent with the diffusion model of Eq. 3.5, our simulation scenarios were extended for longer time peri-

ods. We calculated the α of the survival probability with maximum likelihood, and compared the simulation results with the theoretical data. Let us fix the duration t , dimension d , and wireless density of servers q in Eq. 3.5. The number of dataholders can be viewed as a binomial random variable with the number of trials the number of initial non-dataholders (N_{ndh}) in the scenario and a probability of success (the probability of acquiring the data) equal to $1 - \phi_t$. We run the simulation scenario for a number of times (e.g., 30). In each iteration i ($i = 1, 2, \dots, 30$), there are X_i dataholders at the end of the run (simulation time t). In this case, the likelihood to acquire the data can be solved analytically. We obtain

$$\phi_t = \frac{\sum_i (N_{ndh} - X_i)}{\sum_i N_{ndh}}. \quad (3.8)$$

The intuitive explanation is that the probability of not becoming a dataholder is the *sample proportion* of non-dataholders over the 30 iterations [105]. Then, maximum likelihood can be used for the estimation of α . Following these steps, in the case of FIS with high transmission power and a duration of 20,000 s, the maximum likelihood estimation of α is equal to 0.0332. We repeated the estimation for the FIS scenario, where there is one server in a square kilometer area, varying only the duration. The average value of α was computed as $\bar{\alpha} = \sum_t \alpha_t$, where α_t is the value of α estimated for a simulation duration $t=2,000$ s, 6,000 s, ..., 20,000 s. Figure 3.13 does not indicate any convergence of α to a specific value. Our conjecture is that this is due to the parameters of this scenario, and particularly, the relatively small area of the terrain compared to the large wireless coverage of the server (radius of 230 m compared to one square kilometer of the terrain). Using Eq. 3.5, $(1 - \phi_t) \times 100\%$ matches our simulation results for the percentage of dataholders at time t for the FIS scheme we described.

To evaluate the goodness-of-fit of the trap model with our simulation data, we computed the coefficient of determination between the average percentage of dataholders in the simulations and the trapping model using $\alpha = \bar{\alpha}$ in Eq. 3.5. The coefficient of determination was found to be 0.9921. Figure 3.14 shows the 95% confidence interval of each FIS simulation scenario and the trapping model. The variance in our simulations is large but the model is within the simulation envelope.

3.8 Discussion

7DS can instantiate a different data access mechanism using either the server-to-client or the peer-to-peer paradigm. This chapter focused on the performance analysis of some simple 7DS schemes, each using a different paradigm (e.g., FIS, MIS, and P-P). Specifically, it analyzed the impact of the density of servers (FIS and MIS schemes) and peers (P-P schemes), their wireless

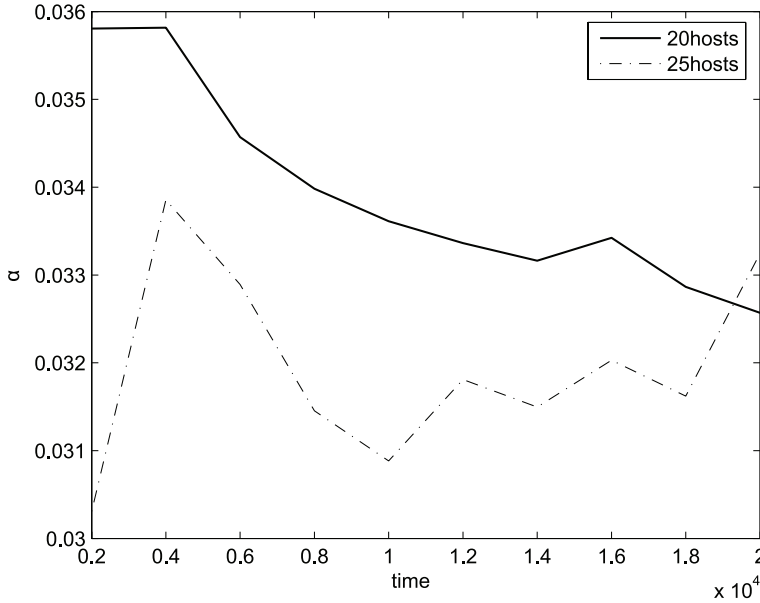


Fig. 3.13. The α parameter of Eq. 3.5, estimated using maximum likelihood for FIS, high transmission power, and varying the tracing period.

range, querying frequency, and energy conservation on the performance of information diffusion.

The server-to-client paradigm in the context of information access for mobile devices has been employed by infostations. A typical infostation is a server that broadcasts data items based on received queries or a predefined schedule. Imielinski and Badrinath were among the first to study the performance of infostations. In their research, they mostly addressed issues related to efficient scheduling algorithms for the server broadcast that minimize the response delay and power consumption of mobile devices and efficiently utilize the bandwidth of the broadcasting channel [198, 303, 79]. Imielinski *et al.* [198] explored methods for accessing broadcast data in such a way that running time (which affects battery life) and access delay (waiting time to receive data) are minimized. The provision of an index- or hash-based access to the data transmitted over the wireless channel can significantly improve the battery utilization. Barbara *et al.* [79] studied a taxonomy of cache invalidation strategies and the impact of clients' disconnection times on their performance.

Assuming a deployment of infostations enabling a wide-area wireless network access, Ye *et al.* [363] evaluated a prefetching operation for mobile users.

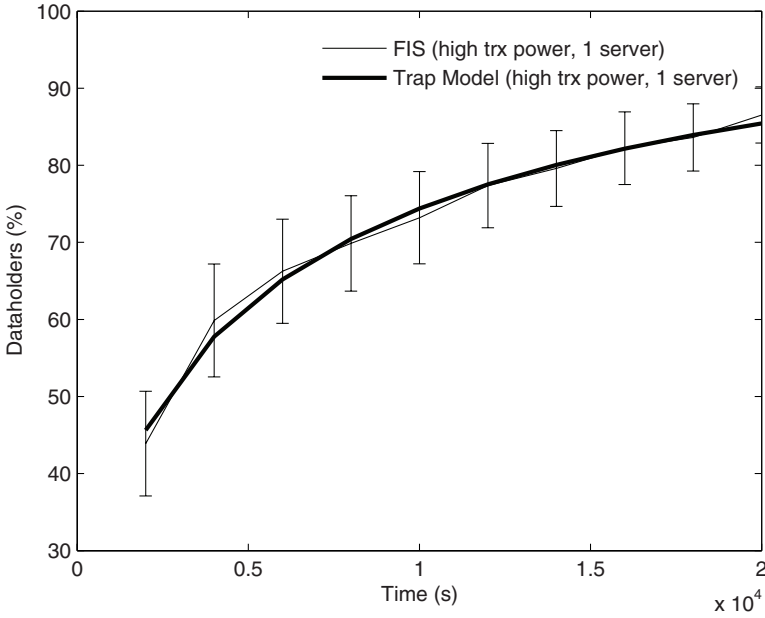


Fig. 3.14. Simulation (FIS) and diffusion model (Trap model) with one server of high transmission power for a longer time horizon (95% confidence interval).

They designed a prefetching algorithm for a map-on-the-move application that exploits a hierarchical representation of information in multiple levels of detail. Based on location, route, and speed information, their algorithm predicts future data access and delivers maps on demand for instantaneous route planning, at the appropriate level of detail. When a mobile device enters the infostation coverage area, it prefetches a fixed amount of bytes that corresponds to a map with a certain level of detail. The effectiveness of infostations was compared to a traditional wide-area wireless network, by varying the infostation density and coverage. Unlike FIS, in which mobile hosts have no wide-area network access, in [363] devices are constantly connected to a low-speed wireless network. Specifically, when these devices are within the infostation coverage, they use a high bandwidth link, whereas, outside these regions, their requests are passed to the server via a conventional cellular base-station. They also showed that it is more efficient to have a larger number of infostations with small range than fewer infostations with large range.

The performance analysis study that is closest to the one presented in this chapter is a followup work by Lindemann and Waldhorst [254]. They modeled the spread of *multiple* data items assuming *finite* buffer capacity at

mobile devices and a least-recently-used buffer scheme. Their analysis explored several variants of 7DS, focusing on their long-run performance. They reported the following interesting results:

- Neither the transmission range nor the selected variant of 7DS has a significant impact on the fraction of dataholders in the long run. However for high transmission ranges, the selected variant of 7DS does have a significant impact on the hit rate. Depending on the 7DS variant and the buffer size, hit rates between 0.48 and 0.92 can be achieved.
- The medium transmission range yields higher hit rates than using aggressive power conservation at a high transmission range.

Recent studies have explored the analytical properties of information dissemination in *constantly-connected* networks that form various topologies, including scale-free and small-world networks (e.g., [61, 227, 248, 41]).¹ As mentioned in Section 3.7.2, theoreticians have been also studying the problem of diffusion and particle kinetics. Recently Kesten and Sidoravicius [221] showed that in the long run, all particles will be concentrated in an area that grows linearly. An attractive feature of the diffusion-controlled process is that it can provide elegant theoretical tools to investigate data dissemination for different network topologies. However, the extension of these research efforts to incorporate parameters, such as the expiration of data objects, buffer size, buffer management policy, type of interaction among devices, cooperation strategy, and time-varying network topologies unfolds several challenges.

To simplify the analysis of data dissemination among mobile peers in DTN environments, this monograph considered that peers communicate via broadcasts and restricted forwardings. More efficient routing protocol could be adopted to facilitate the communication among peers. In general, routing, epidemic, and gossiping algorithms for ad-hoc and sensor networks have received a lot of attention since 1980s and numerous routing protocols have been proposed (e.g., AODV, DSR, TORA, DSDV, ADV, ZRP, LAR). To evaluate their performance, comparative analysis studies have been performed, mostly via simulations (e.g., [101, 207, 126, 193, 89, 298, 161, 228]) using metrics, such as the energy dissipation, packet latency, routing overhead, and throughput per flow. However, traditional routing protocols for ad-hoc networks do not perform well in DTNs, since mobile peer-to-peer computing applications often form sparse, intermittently-connected networks with frequently unstable paths. Flooding algorithms can be also problematic in DTNs of mobile devices with limited resources. Despite their simplicity, robustness, and relatively low delay, their high energy, bandwidth, and memory requirements dissuade

¹ Small-world networks are characterized by a high degree of clustering and small distances between any two nodes in the network. Scale-free networks exhibit a power law degree distribution. A power law has a heavy tail, which makes values far beyond the mean much more likely than for light-tailed distributions [166]. Unlike other distributions, such as the exponential distribution, which drops off very quickly beyond the mean, power laws do not possess a characteristic scale.

their usage in such networks. Since early 2000s, several routing approaches for DTNs have been proposed, investigating the following parameters:

- caching policy, buffer size and management (e.g., [127, 254])
- use and control of relaying nodes (e.g., [368, 167, 367, 361, 252]) and relaying policy (e.g., [97])
- use of knowledge about device mobility and location (e.g., [97, 252, 202, 361, 367])

The impact of mobility on the design of forwarding algorithms for DTNs has also generated a lot of interest in the last few years (e.g., [108, 189, 270]). For example, Chaintreau *et al.* [108] analyzed the inter-contact times using traces from real-life testbeds with human mobility and evaluated its impact on forwarding algorithms. The use of forwarding to enhance the information access in ad-hoc networks has been also studied theoretically. An influential paper on the capacity of *static* ad-hoc networks impelled further theoretical studies, some of them analyzing the use of relaying peers to improve the capacity in ad-hoc networks. Specifically, Gupta and Kumar [170] proved that the average available throughput per node is inversely proportional to the square root of the number of nodes in a static ad-hoc network. Equivalently, the total network capacity increases at most as the square root of the number of nodes. Extending these results, Grossglauser and Tse [167] showed that the capacity of an ad-hoc wireless multi-hop network can be enhanced by exploiting forwarding, in which a sender may forward its message further to a mobile relay node. They evaluated the average per-session throughput and its asymptotic performance in such multi-hop ad-hoc networks and showed that the average throughput per source-destination pair of nodes can be kept constant by increasing the density of nodes. However, the delay of a packet may also increase substantially. To provide guarantees on delay, Bansal and Liu proposed a routing algorithm that exploits the mobility patterns of devices, achieving a throughput that is only a poly-logarithmic factor from the optimal [78].

Mobile peer-to-peer systems and routing protocols have been evaluated mostly via simulations. Analyzing their performance under more realistic conditions with respect to user access (e.g., co-residency times of peers, inter-contact times, arrivals in the range of information servers), traffic patterns link conditions, and network topology can reveal important aspects of their performance. The need for such models motivated the research presented in the following chapters.

Empirically-based measurements on wireless demand

This chapter presents our measurements of caching, access, and traffic demand in wireless networks. It describes the wireless infrastructure and data acquisition methodology. Then, it provides an overview of the traffic demand at APs in two major campus-wide wireless infrastructures and presents an application-based characterization of traffic. It explores the spatial and temporal locality phenomena of the wireless information access and evaluates the impact of caching paradigms. Finally, it discusses related work and main conclusions.

4.1 Introduction

IEEE802.11 networks have been rapidly deployed to provide wireless Internet access, especially in universities, corporations, and metropolitan areas. Empirical and performance analysis studies indicate dramatically low performance of real-time constrained applications over wireless LANs (such as [62] on VoIP), and large handoff delays [310, 264]. Moreover, mobile users still experience frequent loss of connectivity and high end-to-end delays when they access the wireless Internet. For example, the overhead of scanning for nearby APs is routinely over 250 ms, far longer than what can be tolerated by highly interactive applications such as voice telephony.

Wireless LANs have more vulnerabilities, bandwidth, and latency constraints than their wired counterparts. It is critical to understand the performance and workload of wireless networks and develop wireless networks that are more robust, easier to manage and scale, and more able to efficiently utilize their scarce resources. While in several cases over-provisioning in wired networks is acceptable, it can become problematic in the wireless domain. A number of mechanisms, such as capacity planning, resource reservation, device adaptation, and load balancing, need to be employed to support such networks. Real-life measurement studies can be particularly beneficial in the development and analysis of such mechanisms, as they can uncover deficiencies

of the wireless technology and different phenomena of the wireless access and the workload. The existence of testbeds, tools, and benchmarks is of tremendous importance. Rich sets of data can impel modeling efforts to produce more realistic models, and thus, enable more meaningful performance analysis studies. Recently there have been several empirical measurement studies on the following issues:

- traffic load [338, 75, 74, 233, 177, 261, 282]
- user access [83, 74, 288, 340, 108, 223, 201, 203]
- handoff [310, 264]
- delay and packet losses in the MAC [134] and TCP connections [180]
- link quality and routing [122, 87, 57]

Measurements on IEEE802.11-based mesh networks have also received a lot of attention [57, 85, 87, 308, 257, 45, 43, 70, 129, 130, 45].

This chapter focuses on the wireless demand in large-scale wireless infrastructures and presents an exploratory analysis of the amount and type of demand. Section 4.2 presents the wireless infrastructure and Section 4.3 describes the data acquisition methodology. The main terminology that will be used in the empirical measurement studies included in this work is defined in Section 4.4. Section 4.5 provides an overview of the workload of APs of two major campus-wide wireless networks, in terms of the number of bytes sent and received, number of packets sent and received, and number of associations and roaming operations. An application-based characterization of the wireless demand is discussed in Section 4.6. Section 4.7 investigates the locality of the web URLs accessed from the wireless infrastructure and evaluates several caching paradigms. Finally, Section 4.8 discusses related research and summarizes our main conclusions and future work plans.

4.2 Campus-wide wireless infrastructure

The UNC began the deployment of its wireless infrastructure in 1999, providing coverage for nearly every building in the 729-acre campus, encompassing a diverse academic environment, which includes university departments, programs, administration, activities, and residential buildings. In these buildings, there are 26,000 students, 3,000 faculty members, and 9,000 staff/administrative personnel [5]. Of the 26,000 students, 61% are undergraduates, and more than 75% own a wireless laptop.

Most of the APs belong to three different series of the Cisco Aironet platform: the state-of-the-art 1200 Series, the widely-deployed 350 Series, and a few older 340 Series. The 1200s and 350s run Cisco IOS, while the 340s run Vx-Works. Since each AP has a unique IP address, we used an AP's IP address to determine its unique AP ID number. Each AP has a coverage area determined by the radio propagation properties around the AP. Each IEEE802.11-enabled device that communicates with the campus wireless network is called a *client*,

is assumed to have a unique MAC address, and is assigned a positive unique ID number based on its MAC address. A client communicates via the network by associating itself with an AP; in this case we say that such a client visits the AP.

The wireless infrastructure has expanded substantially during the last few years. Table 4.1 shows the evolution of the wireless infrastructure and the significant increase of APs and wireless clients.

Tracing period	Clients	APs
February 10–April 27, 2003	7,694	232
17–24, October 2004	8,880	459
2–9, March 2005	9,049	532
13–20, April 2005	9,881	574
September 29–November, 2005	14,712	574

Table 4.1. The evolution of the wireless infrastructure at UNC in terms of number of APs and wireless clients.

4.3 Monitoring and data acquisition

We monitored the wireless infrastructure at UNC and collected extensive wireless traces, such as packet headers, SYSLOG, SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP), TCP flow, and signal strength-based data.

Monitoring large-scale wireless networks comes with several challenges. Often monitoring tools are limited in their capabilities because they cannot capture all relevant information due to either hardware limitations, the proprietary nature of hardware and software, or hidden terminals. Furthermore, the implementation of many protocol features of IEEE802.11—such as the rate adaptation and transmission power control—are vendor-specific and their details are not publicly available. At the same time, wireless measurements feature high complexity due to transient phenomena, missing values, and spatio-temporal dependencies. Transient phenomena are due to roaming and radio propagation issues, while failures of the monitoring devices and APs, lost UDP packets of SYSLOG events or other measurement messages result in missing values in data traces. It is a non-trivial task to monitor areas of intermittent connectivity and select the physical and network position of monitors in large-scale infrastructures. The type of phenomena that needs to be studied determines the amount of traffic at multiple locations that needs to be captured, its resolution, and the correlation among multiple sources of data that needs to be performed.

The next paragraphs describe the traces and the main terminology used in the measurement-based studies of this work.

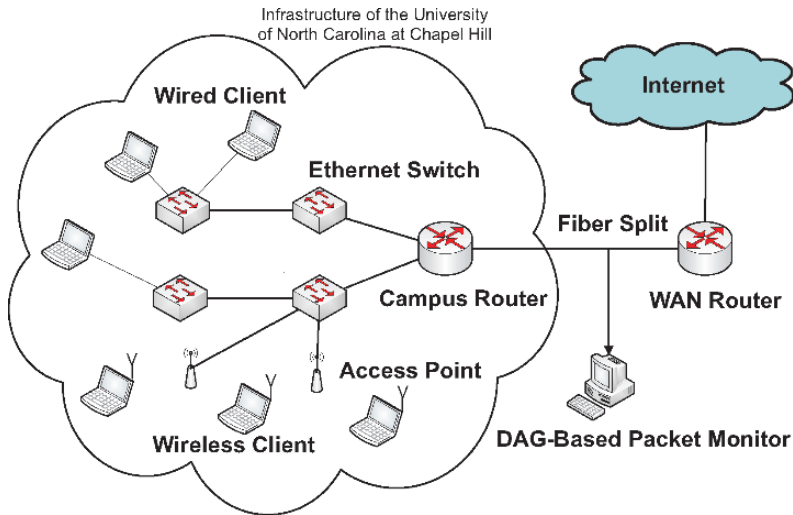


Fig. 4.1. The campus-wide wireless infrastructure and packet monitor tool.

4.3.1 Packet header traces

The bulk of the campus wireless network has a single aggregation point that connects to a gateway router. This router provides connectivity between the wireless network and the wired links, including all of the campus computing infrastructure and the Internet. Packet header traces were collected with a high-precision DAG-based monitoring card (Endace 4.3GE). The card was installed in a high-end FreeBSD server and captured all packets traversing the link between UNC and the Internet in both directions (Figure 4.1). The collected packet traces do *not* include the “internal” wireless traffic (i.e., traffic generated between wireless clients at UNC). In general, monitoring high-speed links with a software-only system may result in inaccuracies. Specifically, the traffic has to be forwarded from the network interface to the monitoring software, using the system bus which may not be fast enough, especially when the monitored link is under heavy traffic load conditions. As a result, the monitor will not record dropped packets. In addition, the buffering that is involved across the different layers—the network interface to the operating system—may result in inaccurate timestamps. DAG is a specialized hardware—has been widely used in network measurement projects—that overcomes these problems. The accuracy of DAG traces can be on the order of nanoseconds [178].

The monitoring period was 178.2 hours in 2005 and 192 hours in 2006, yielding 175GB and 365GB of packet headers, respectively. The sharp in-

crease in the trace size indicates the significant growth of the wireless demand between these periods.

4.3.2 HTTP traces

The HTTP traces were based on packet headers collected from the FreeBSD monitoring system described in the previous section. The tracing tool TCP-DUMP was employed to collect all TCP packets with payloads that begin with the ASCII string “GET” followed by a space. The full frame was collected as a potential HTTP request. We did not restrict our collection to the standard HTTP port, allowing us to record HTTP requests sent to servers on non-standard ports, which include many common peer-to-peer file-sharing applications. The packet trace was then processed to extract the HTTP GET requests contained therein.

From each packet, the following information was recorded:

- *time* of the packet’s receipt with one-second resolution
- *hostname* specified in the request’s Host header
- *Request-URI*
- *hardware MAC address* of the IEEE802.11 client

If all of these items were not available in a packet, that packet was not included in the recorded requests. Using these criteria, 8,358,048 requests for 2,437,736 unique URLs were traced and included in the analysis. By recording the traffic before it had passed through an IP router, we were able to capture the original MAC header—as generated by the IEEE802.11 clients—for transmission to the gateway router.

The HTTP traces were collected during the tracing period between February 26 and March 24, 2003. During that period, the campus used primarily Cisco Aironet 350 802.11 APs, although some areas of the campus were serviced by older APs from other manufacturers. As the SYSLOG traces indicated, the infrastructure was accessed by 7,694 distinct wireless clients and 37% of them made one or more HTTP requests during that period.

4.3.3 SNMP traces

SNMP is one of the most widely available monitoring services. Every AP on the market supports monitoring using SNMP, so it is important to understand how much operators and researchers can learn from SNMP data.

For the comparative study of the workload of wireless campus-wide networks, SNMP data was acquired using a non-blocking SNMP library for polling every AP precisely every five minutes in an independent manner. This eliminated any extra delays due to the slow processing of SNMP polls by some of the slower APs.

4.3.4 SYSLOG traces

The majority of APs on campus were configured to send trace data to a SYSLOG server in our department. There are seven types of events that trigger an AP to transmit a SYSLOG message. These messages and their corresponding events are interpreted as follows:

AUTHENTICATED: A card must authenticate itself before using the network. Since a card still has to associate with an AP before sending and receiving data, we ignored any authenticated messages.

ASSOCIATED: After it authenticates itself, a card associates with an AP. Any data transmitted to and from the network is transmitted by that AP.

REASSOCIATED: A card may reassociate itself with a new AP (usually due to higher signal strength) or the current AP. After a reassociation with an AP, any data transmitted to and from the network is transmitted by that AP.

ROAMED: After a reassociation occurs, the old AP and sometimes the AP with which the card has just reassociated send a roamed message. Since we still receive the reassociated message, we can ignore this message as well.

RESET: When a card's connection is reset, a reset message is sent.

DISSOCIATED: When a card wishes to disconnect from the AP, it disassociates itself. We ignore any disassociated messages from a card if the previous message for that card was a disassociated or deauthenticated message.

DEAUTHENTICATED: When a card is no longer part of the network, a deauthenticated message is sent. It is not unusual to see repeated deauthenticated messages for the same card, with no other type of events for that card in between. We ignore any deauthenticated messages for a card if the previous message for that card was a disassociated or a deauthenticated message. A disconnection message describes either a disassociated or deauthenticated message.

The collection of the packet header, HTTP, SNMP, and SYSLOG traces is described in detail in [51].

4.3.5 Privacy assurances

To avoid disclosure of the identity of individual users and of the sites that a user has been visiting, we stored and used SHA1 hashes of the client's MAC address, request hostname, and requested path of the HTTP requests.

4.3.6 Client identification

The MAC address uniquely identifies an IEEE802.11-enabled device and is assumed to be coupled to a specific computer.

4.4 State, history, visits and sessions

Using the associated, reassociated, deauthenticated, and disassociated SYSLOG events, the following structures were defined to characterize the access pattern. Note that we assumed that each event occurs at the time of the timestamp in the corresponding SYSLOG entry.¹

State: A state represents the AP with which a client is currently associated. When a client is connected to the network, its state is the numeric ID of the AP with which it is currently associated (via an association or a reassociation). When the client is disconnected from the network, its state is defined to be “0”. Since we do not know where the clients are before the trace begins, each client is considered to be in state 0 at the beginning of the trace. We now define these structures:

State history: The state history of a client is the ordered sequence of *states* that the client has visited.

Reconnection threshold: Sometimes a client will disassociate or deauthenticate for a single second and then associate or reassociate. We found that a user was disconnected 71,988 times for one second or less and 104,763 times for 30 seconds or less (and reconnected after that). Whenever a client is disconnected for one second or less, we do not consider the client to have disconnected from or left the network, but instead to be in the middle of a reconnection process. We decided to use one second because it accounted for such a large percentage of all the times such short periods of disconnection occurred. We believe that this represents more accurately the user’s intentions. These rules left us with 2,474,394 useful SYSLOG events for 6,186 clients (Table 4.2) and allowed us to define the following terms:

Visit: A client begins a *visit* to an AP when a (re)association message is received from that AP for that client and ends that visit when *any* message from *any* AP is received for that client. The difference in the timestamp of these two messages defines the duration of the visit. Each visit is “associated” with a state.

Session: A session is a sequence of visits to APs and used to capture an episode of a *continuous* wireless access to the infrastructure. A session begins when a currently disconnected client associates with the infrastructure and ends when the next disconnection message is received. The difference in the timestamps between the disconnection message and the first (re)association message defines the duration of the session. Two sessions of the same client cannot be overlapping in time. A session can be *mobile*, *roaming*, or *stationary*.

¹ The exception is that if a client is deauthenticated due to an inactivity period of thirty minutes (or more), the disconnection was considered to have occurred thirty minutes before the timestamp that appears in the corresponding deauthenticated SYSLOG entry. The inactivity period of thirty minutes is a default value for most of the clients in our infrastructure.

Event type	Events	Clients	APs	Buildings
Total syslog	8,158,341	7,694	222	79
Useful syslog	2,474,394	6,186	222	79

Table 4.2. Summary of SYSLOG statistics.

Inter-AP transition: If a client is currently associated to an AP, an inter-AP transition is defined as a (re)association to a *different* AP. The two APs may or may not be in the same building.

Inter-building transition: If a client is currently associated to an AP at a certain building, an inter-building transition is defined as a (re)association to an AP located in a different building.

Roaming session: A roaming session is a sequence of *consecutive* visits to two or more *distinct* APs.

Mobile session: A mobile session is a special type of roaming session that comprises a sequence of consecutive visits to two or more APs located in *different* buildings.

Roaming (mobile) client: A client with a roaming (mobile) session is called a roaming (mobile) client.

Drop-in client: A drop-in client is a card that visits two or more buildings in the period of time in question. Drop-in clients may have disconnections in between the visits to these buildings.

4.5 Wireless traffic demand at APs

Measurement studies indicate that several hotspot APs in campus-wide environments exhibit diurnal and weekly periodicities in their traffic load [177, 75, 287, 282]. This section examines the amount of traffic of APs, bytes and packets, and number of association and roaming operations. It also presents a comparative system-wide analysis that provides a useful view of the entire utilization of two large-scale wireless networks from the perspective of APs.

4.5.1 Data acquisition

SNMP data from the wireless infrastructures of UNC and Dartmouth was collected. The UNC dataset was collected between 9:09 AM, September 29th, 2004 and 12 AM, November 25th, 2004. The Dartmouth trace corresponds to the dataset studied in [177] and was acquired using a similar approach. It was collected between November 1st, 2003, and February 28, 2004, thus the duration of this trace is twice the duration of the UNC one. This trace includes 6,875 unique MAC addresses which were associated with one or more APs during the data collection period. This number is larger for the UNC trace, which reports on the activity of 14,712 unique MAC address. Thus, while the number of APs in both campus networks is similar, there are twice as many wireless clients in the UNC trace.

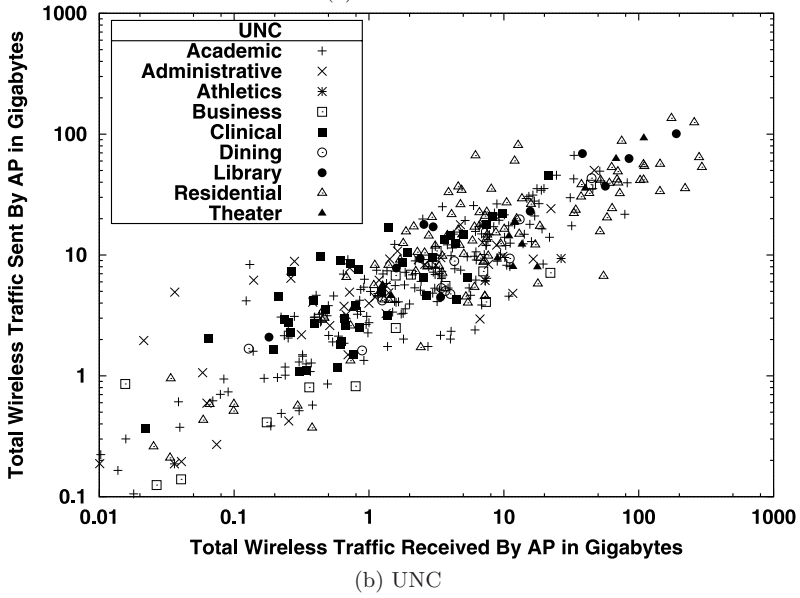
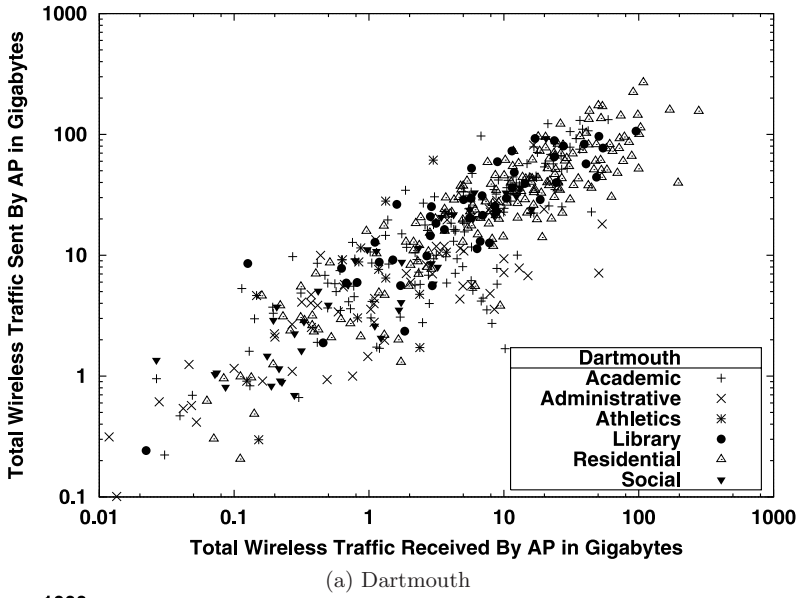
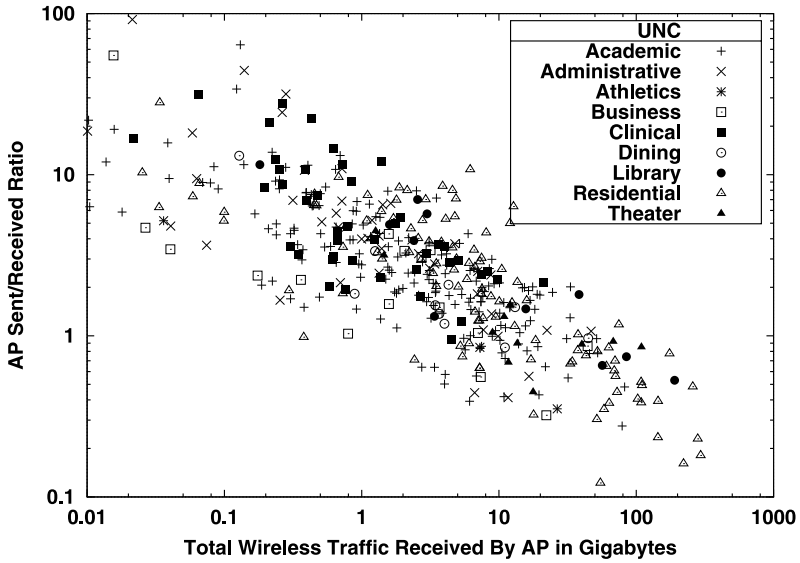
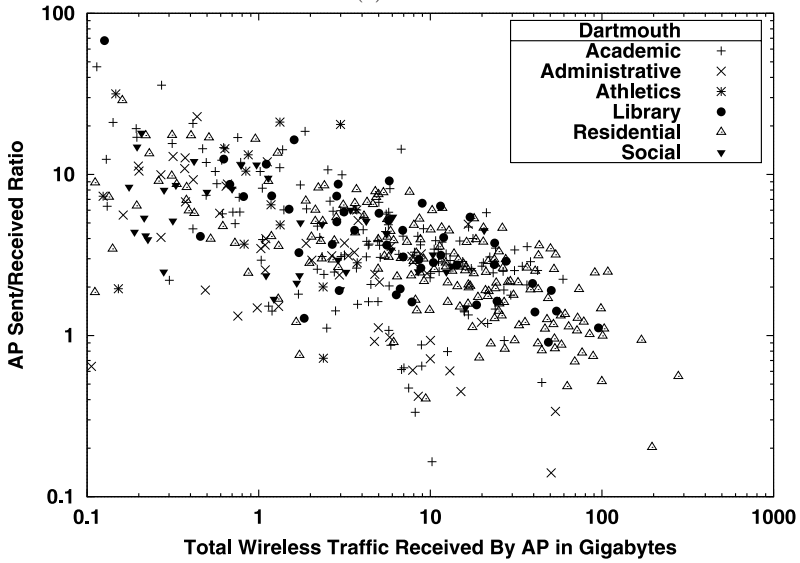


Fig. 4.2. Total wireless traffic sent and received per AP (by building type).



(a) UNC



(b) Dartmouth

Fig. 4.3. Ratio of sent to received traffic compared to received by AP.

4.5.2 Comparative analysis of wireless traffic load at APs

A surprising degree of similarity in the characteristics of the UNC and Dartmouth wireless demand was found. Our results therefore provide strong evidence in support of the development of parsimonious workload models of campus wireless networks.

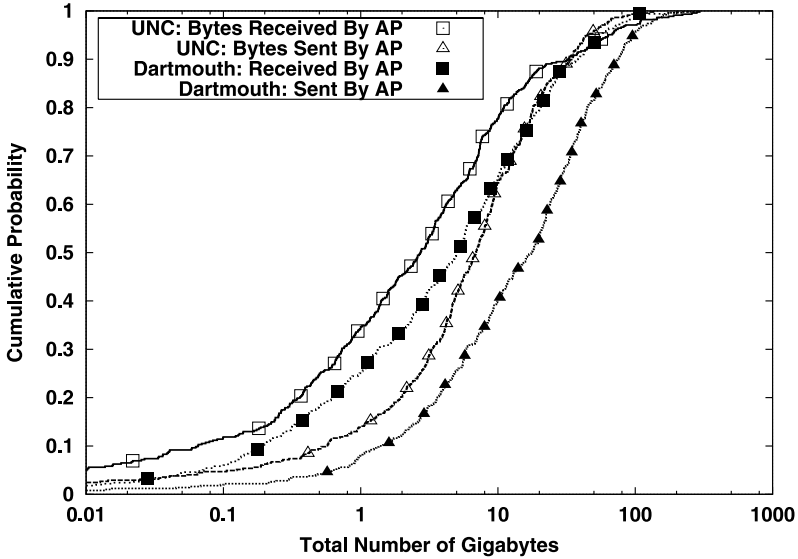


Fig. 4.4. Total amount of traffic transferred at UNC and Dartmouth in each direction.

Specifically, our analysis reveals the following:

- There is a wide range of workloads and that log normality is prevalent in both the UNC and Dartmouth traces.
- In general, the traffic load in both wireless infrastructures is light, although there are long tails (Figures 4.4 and 4.6).
- No clear dependency with the type of building at which the AP is located exists, although some stochastic ordering is present in the tail of the distributions.
- An interesting dichotomy among APs is prominent in both of the infrastructures: APs dominated by uploaders and APs dominated by downloaders (Figure 4.5). Specifically, we observed that as the total wireless traffic received at an AP increases, there is also an increase in its total traffic sent (Figure 4.2) and, a simultaneous decrease in the sent-to-received ratio (Figure 4.3).

- The number of non-unicast wireless packets is substantial. Furthermore, the number of unicast received packets is strongly correlated in the log-log scale with the number of unicast sent packets (Figure 4.7 (a)).
- While the majority of APs send and receive packets of relatively small size, a significant number of APs show rather asymmetric packet sizes, *i.e.*, APs with large sent and small receive packets, and APs with small sent and large receive packets (Figure 4.8).
- The distribution of the associations and roaming operations was found to be quite heavy-tailed.
- There is a correlation between the traffic load and number of associations in the log-log scale (Figure 4.9).

4.6 Application-based characterization of wireless demand

As the wireless user population increases, characterization of its workload can facilitate more efficient network management and better utilization of users' scarce resources. While there have been several studies looking at the application cross-section in wired networks [333, 341, 169, 109], such attempts are limited in the case of wireless networks [177].

Using the port number to classify flows may lead to significant amounts of misclassified traffic due to dynamic port usage, overlapping port ranges, and traffic masquerading. Often, peer-to-peer and streaming applications use dynamic ports to communicate, and even worse, the port ranges of different applications may overlap. Furthermore, several malware or peer-to-peer applications may try to masquerade their traffic under well-known "non-suspicious" ports, such as port 80. Besides the well-documented limitations of application identification [216, 268, 215] inherent additional complications in wireless networks, such as the increasing overheads of data collection due to the need of multiple monitoring points, cross-correlation of different type of traces, and transient phenomena due to the radio propagation and mobility, have led the community to assume that the expected workload of wireless networks follows the general trends of Internet applications.

To avoid this "known-port limitation" [268, 215], we employed the BLINC tool [216] which performs classification of flows into applications based on the transport-layer footprint of the various application types.

For the application-based classification study, we processed packet-header traces collected at one of the access routers at UNC and client-based SNMP data from all APs. The SNMP data was used to associate each flow with the corresponding MAC address and AP information. Approximately 9,125 distinct internal IPs which were mapped to approximately 3,241 unique MAC addresses were observed in the traces. BLINC was able to classify 86% of our flows into application types. Some cases of misclassifications were due to outlying user

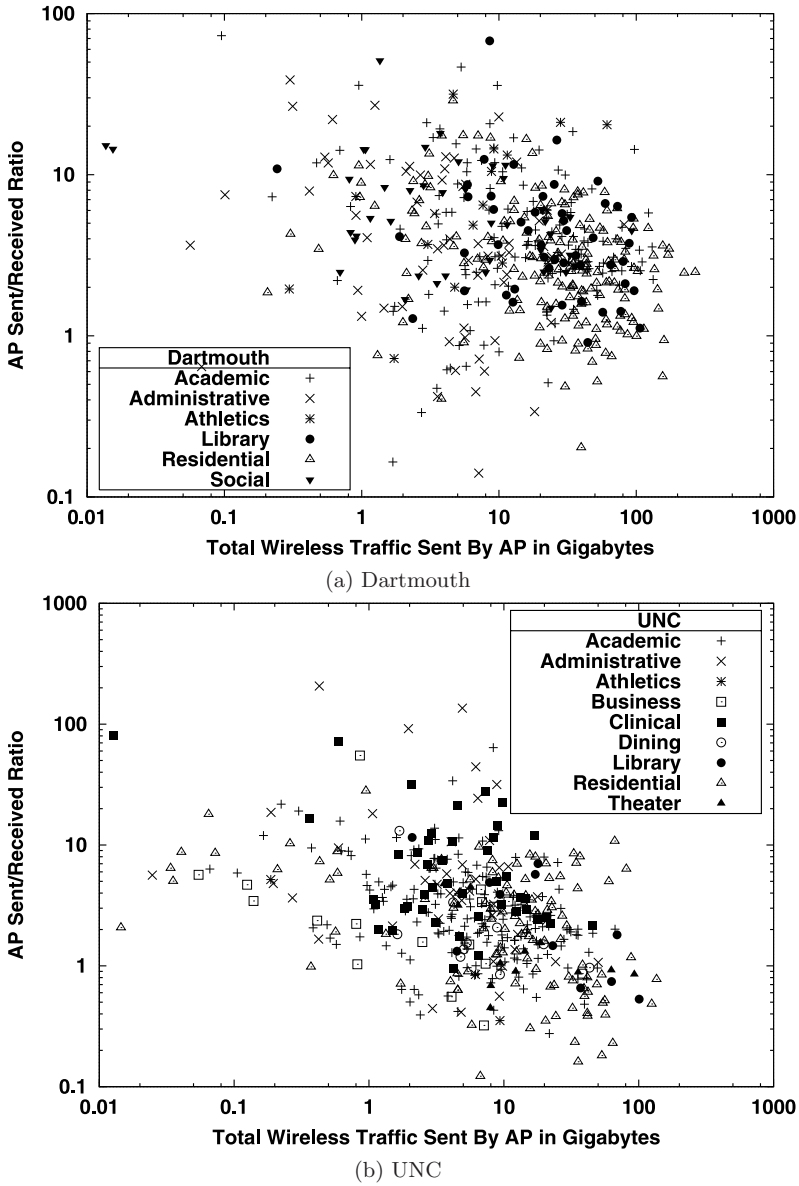


Fig. 4.5. Ratio of total traffic sent and received compared to total traffic sent per AP.

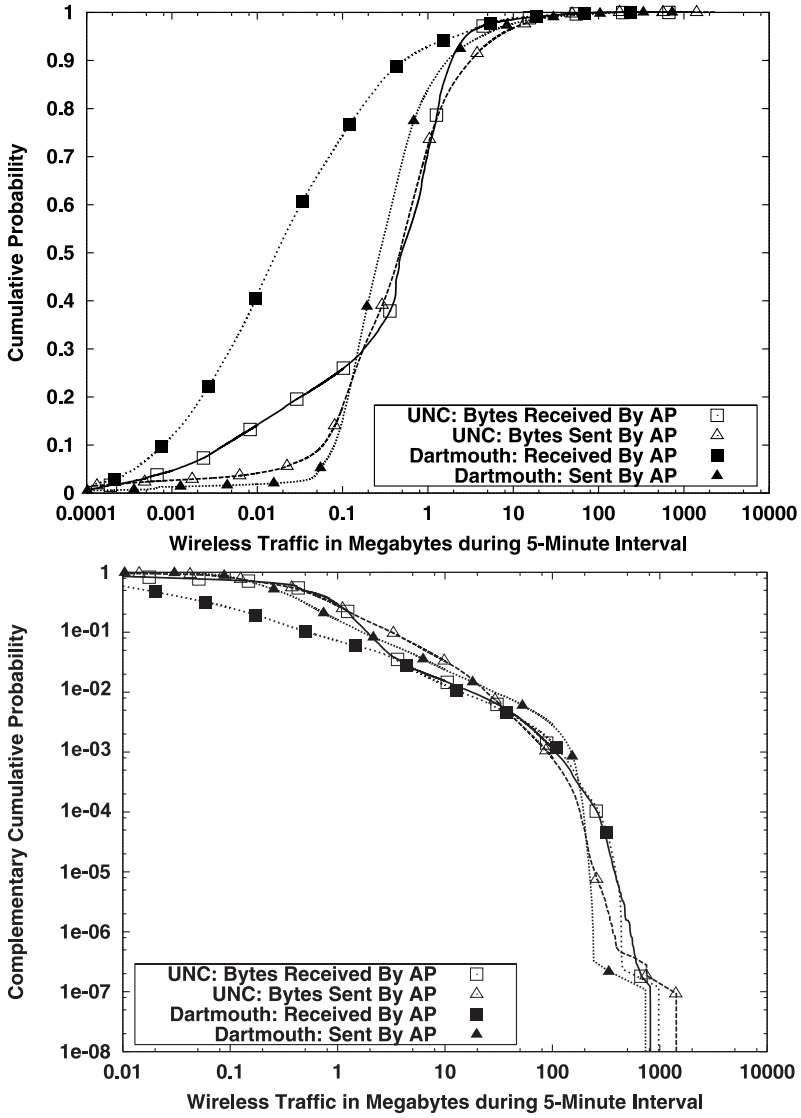
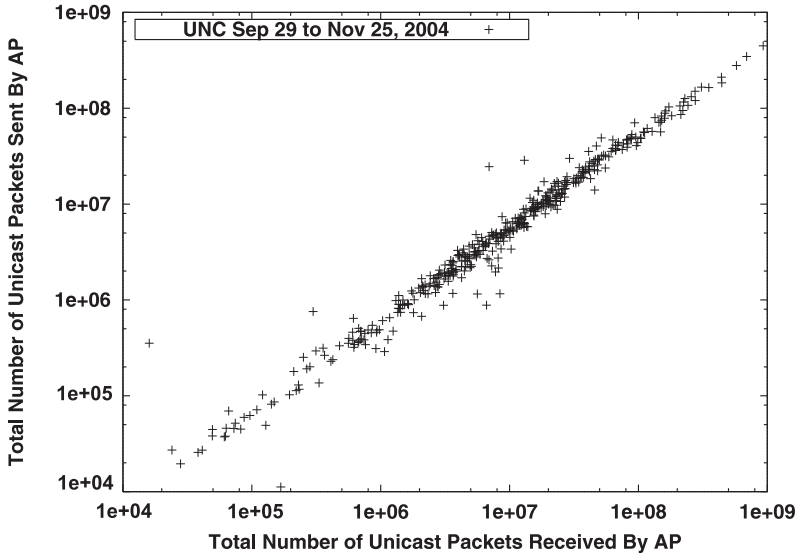
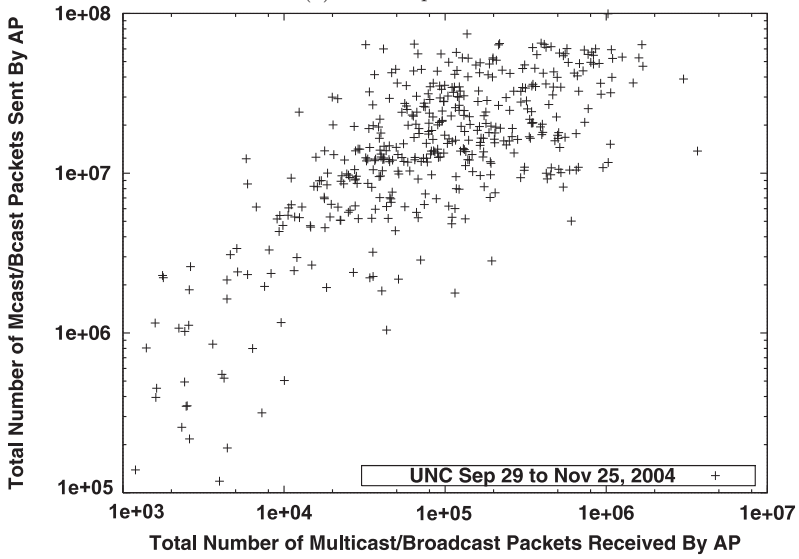


Fig. 4.6. Total amount of traffic transferred during 5-minute interval.



(a) Unicast packets



(b) Multicast/Broadcast packets

Fig. 4.7. Total number of packets sent and received by an AP at UNC.

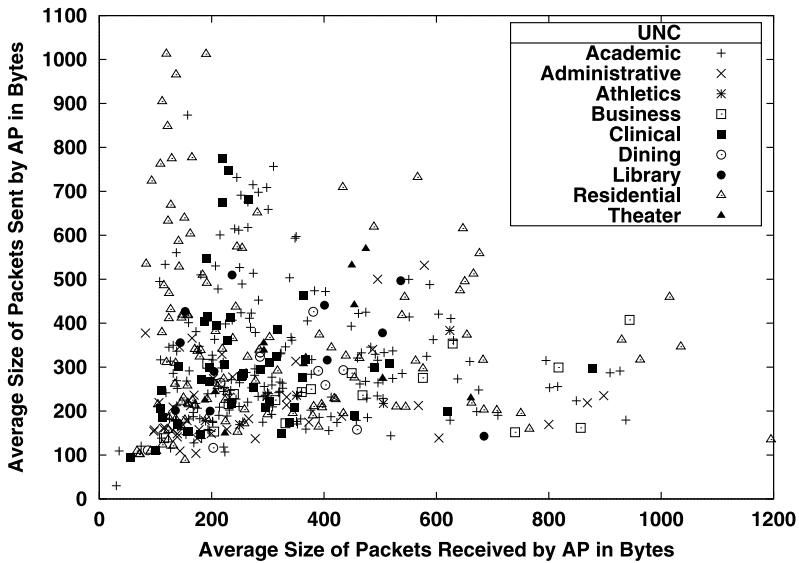


Fig. 4.8. Average size of packets sent and received by an AP at UNC in bytes.

behavior. Nearly 5% of the users were responsible for 98% of misclassified web traffic and thus all these flows were excluded. Our main results are summarized as follows;

- The most popular applications are web browsing and peer-to-peer, accounting approximately for 81% of the total traffic. Most users are also dominated by these two applications.
- Network management and scanning activity are responsible for 17% of the total flows.
- While building-aggregated traffic application usage patterns appear similar, the application cross-section varies within APs of the same building.
- Most wireless clients appear to use the wireless network for one specific application that dominates their traffic share.
- File transfer flows, such as FTP and peer-to-peer, are heavier in the wired network than in the wireless one.
- The traffic share across applications is significantly affected when clients associate with new APs. This appears to be independent of the specific application type.
- There is a dichotomy among APs, in terms of their dominant application type and downloading and uploading behavior.

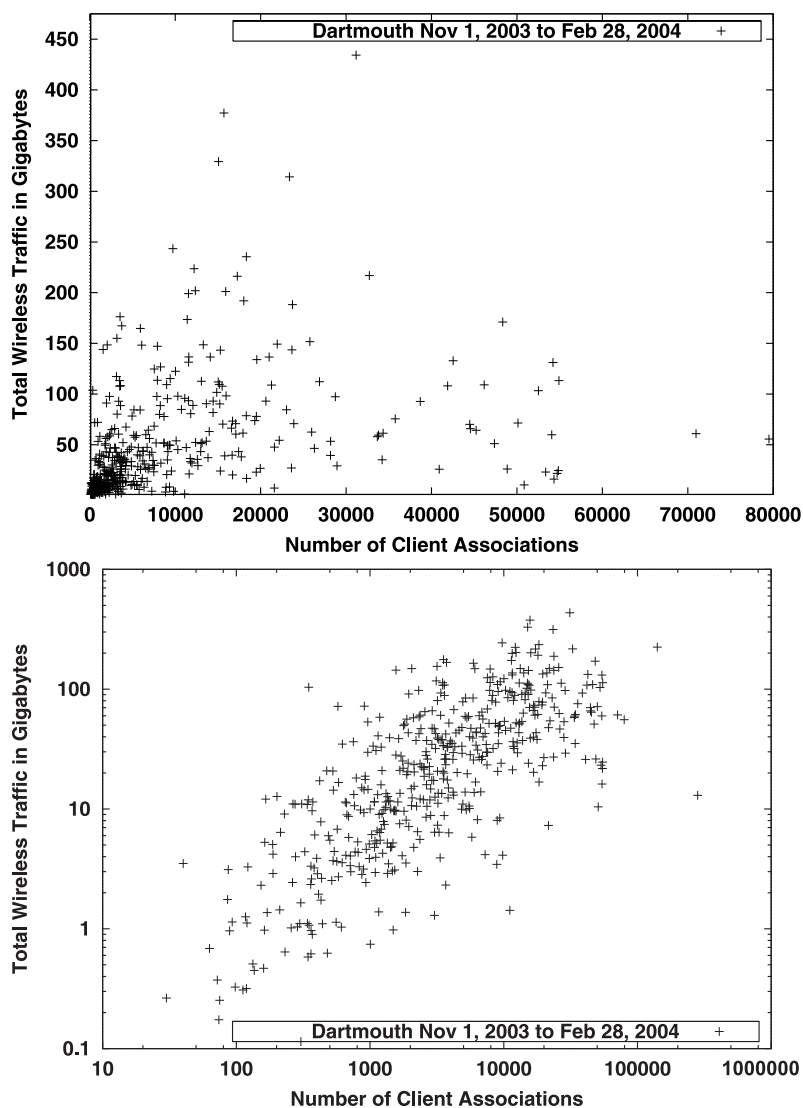


Fig. 4.9. Total wireless traffic and number of client associations at Dartmouth.

As new wireless applications and services are deployed—reshaping the wireless arena—it would be interesting to observe and analyze the evolution of the wireless access in the spatial and temporal domain.

4.7 Locality of web objects

The peer-to-peer paradigm exploits the *spatial locality of queries and information access*. Chapter 3 showed via simulations that in settings with high spatial locality of information and frequent disconnections from the Internet, these peer-to-peer systems can enhance information access by reducing the average delay in receiving the data. Empirical studies in wireless networks have indicated that web and peer-to-peer applications are among the most prominent type of access [300, 234, 85].

This section examines the spatio-temporal characteristics of the wireless access through measurements. Does the information access in wireless production networks exhibit spatial locality? How effective can different caching schemes be and what would be the impact of the peer-to-peer paradigm in such networks? Although the web is not primarily a location-dependent or collaborative application, its prevalence motivated us to start our analysis by focusing on web requests in a large-scale wireless network.

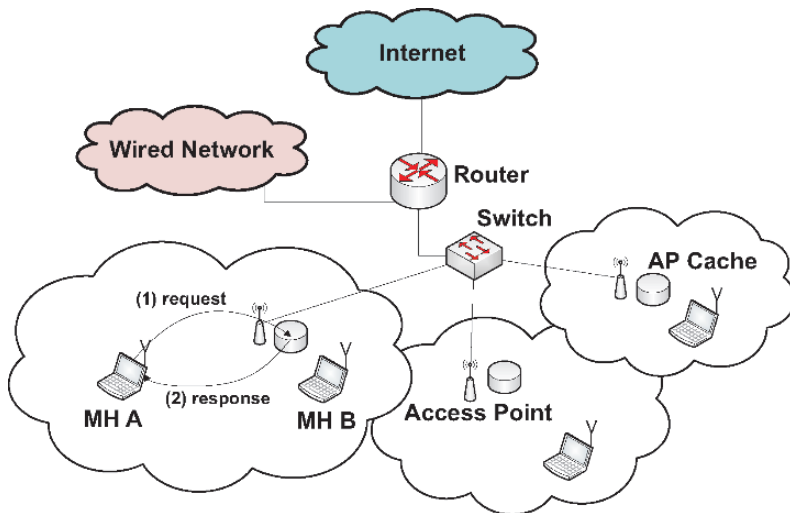


Fig. 4.10. AP cache: Devices request and acquire data from the cache of their local AP.

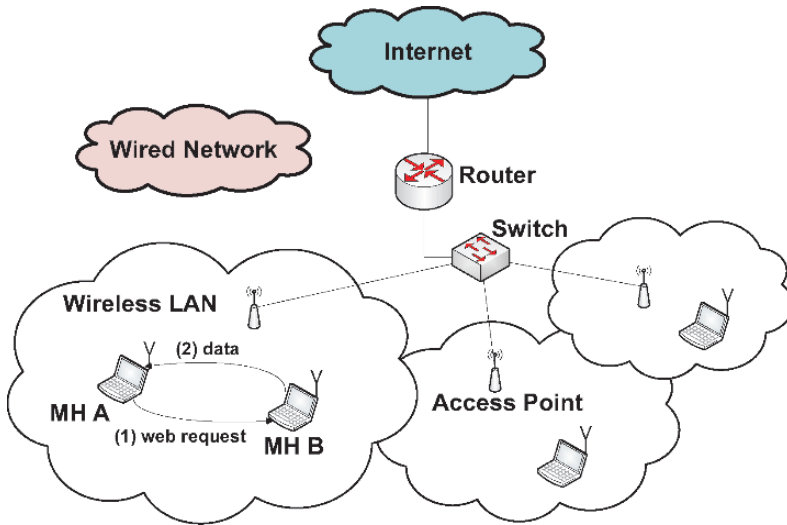


Fig. 4.11. Peer-to-peer caching: Devices request and acquire data from the caches of peers associated with the same AP.

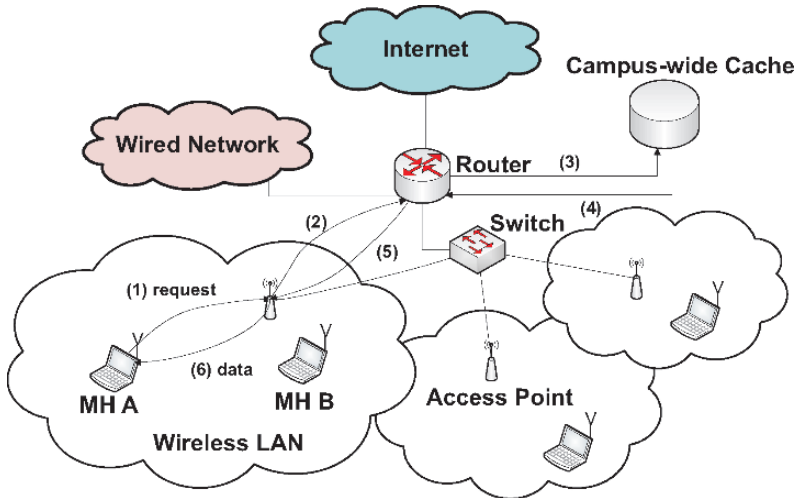


Fig. 4.12. Campus-wide cache: Devices request and acquire data from a campus-wide cache.

The temporal locality identifies the frequency and temporal aspects of repeated requests for certain information. The spatial locality focuses on the AP and building in which a repeated request occurs and indicates if the repeated request originated from a nearby client, a client within the same AP, or a client in the same building.

Three main caching paradigms are explored: *user cache*, *cache attached to an AP or a building*, and *peer-to-peer caching*. For the evaluation of these paradigms, the following assumptions were made:

- A user cache is considered to be the web browser cache.
- A cache attached to an AP or a building will serve the wireless clients associated to that AP or to APs of that building, respectively.
- In peer-to-peer caching, clients associated with the same AP act as cooperative caches for each other.

Figures 4.10, 4.11, and 4.12 illustrate the AP cache, peer-to-peer caching, and campus-wide cache paradigms, respectively.

Web requests may exhibit different locality characteristics and can be classified into the following categories:

- *same-client*
- *same-AP*
- *AP-coresident-client*
- *same-building*
- *campus-wide*

This classification is hybrid in that it exhibits both temporal and spatial characteristics. The following sections discuss these characteristics and present the locality characteristics of a large campus-wide wireless infrastructure.

4.7.1 HTTP requests model

Two requests are considered to be from *the same client*, if they were generated by clients that have the same hashed MAC address, and two requests are considered to be for the *same URL*, if they have the same hashed hostname and request path.

A post-processing phase was performed that conceptually examined every request in the HTTP trace and identified the AP via which it was made using the SYSLOG trace [115].

4.7.2 Same-client repeated requests

A same-client repeated request occurs when a single client requests an object that it has requested in the past. The cause could be any of the following:

Subsequent request: A client intentionally requests an object that it has requested in the past but had not been satisfied by the browser cache subsequent request. Such a request would represent genuine ongoing interest by that client.

Client reloads: A client reloads a page. This may occur when the page has not been transmitted properly or to refresh content (e.g., live sports scores).

Automatic reloads: Many popular pages (such as headline-news and weather sites) cause the browser to re-load the page periodically. While the page is displayed, the browser will periodically re-request it. Some of these requests could also be considered indicative of continued interest by the client.

Packet retransmissions: If the first packet containing the request was not known by the client to have reached its destination, TCP specifies that the client retransmit the packet. Both requests are distinct requests. However, such retransmissions are expected to be rare [255].

This study is subject to the effects of browser caching; if the requested object is in the browser's cache, then no HTTP request will be generated. Some, but not all, browsers follow HTTP's specification for determining the freshness of a cached object. Also, we speculate that a percentage of the repeated requests are conditional HTTP GET requests. This measure does not account for the location of the client and therefore reveals temporal but not spatial locality. The temporal locality of these requests was computed as follows: for each request in the trace, we searched for previous references to the same URL made by that same client. If such request was found, we recorded the time elapsed since this request occurred.

4.7.3 Same-AP repeated requests

When an object is requested multiple times within the same AP's range, those are called same-AP repeated requests. This measure does not account for the client that makes the request; i.e., the repetition can occur due to a single client or several clients requesting the same object within a single AP's range.

4.7.4 AP-coresident-client repeated requests

A central question for motivating information sharing systems targeting mobile users is the following: How often are users who are interested in the same things near one another? To answer this question, object and client-AP, as well as, object and client-building correlations were examined. These spatial locality properties of wireless web access can impact caching.

An AP-coresident client repeated request is said to occur when a client in an AP's area requests an object that has been requested at some time in the past by another client who is in the same AP's area at the time that the new request is made. Note that the other client which requested the object in the past may have requested the object while at a different location.

For each request in the trace, we searched backwards in time for previous references to the same object made by a *different* client that is *currently associated with the same* AP. If such a request was found, the time that has elapsed since this request was recorded. Figure 4.13 displays the fraction of same-client, same-AP, and AP-coresident client repeated requests

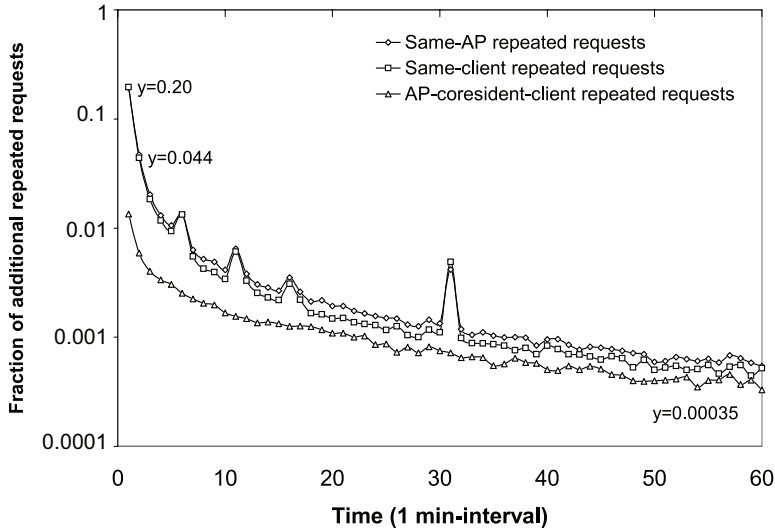


Fig. 4.13. Fraction of additional repeated requests within a one-hour interval. The number of requests considered is at least 7.6 million. Over 2,800 clients are represented.

respectively, for an interval equal to one hour. More specifically, the fraction of repeated requests at each minute is equal to the *additional* repeated requests that occur in that minute of the first hour. For example, within the first minute the fraction of repeated requests is at least 0.19 for the same-client, same-AP, and 0.01 for AP-coresident client. In the second minute, an *additional* 0.04 fraction of requests are same-client and same-AP repeated requests, and the fraction of *additional* repeated requests is 0.006 for AP-coresident client repeated requests.

Same-client and same-AP repeated requests exhibit some five-, ten-, fifteen-, and thirty-minute periodicities. Furthermore, the fraction of repeated requests for same-client and same-AP is similar and higher than that of AP-coresident repeated requests. As many as 37% of all requests would be unnecessary if every object on the web had a cache lifetime of at least an hour. This indicates the impact of the client's web browser cache, assuming that all browsers observe the HTTP standard for caching.

The repeated requests follow a power law with exponential coefficients of -1.31, -1.27, and -0.84 for same-client, same-AP, and AP-coresident client, respectively. The coefficient of determination is at least 0.94 for all of them. These coefficients indicate that the temporal locality is more apparent in the same-client but not in the AP-coresident client caches.

The web requests exhibit a strong temporal locality highlighted by the decreasing trend that becomes more prominent for larger time intervals. As shown in Figure 4.14, within a day the percentage of repeated requests is 44% for same-client, 48% for same-AP repeated requests, and only 14% for AP-coresident client repeated requests. On the second day, an *additional* fraction 0.02 of requests are same-client and same-AP repeated requests, and the fraction of repeated requests is 0.02 for AP-coresident client repeated requests.

Our cache hierarchy consists of the client cache at the lower level of the hierarchy, caches at APs, caches of co-resident peers, and a campus-wide cache. Figure 4.15 focuses on the impact of each caching paradigm when there is a miss in the other caches of the cache hierarchy. For example, it shows the impact of the caches at APs for requests that cannot be served by the client cache (“Same-AP \cap \neg Same-client repeated requests”) as well as the impact of the peer cache for requests that cannot be served by the client cache or the cache of the local AP.

The hit ratio results are conservative, because they include compulsory (cold start) misses. This effect is reduced by taking measurement traces over 26 days. On the other hand, we assumed infinite cache size and that shared documents are cacheable, thus the following hit ratios are *ideal* hit ratios. A cache at each AP would achieve an ideal hit ratio of 55% for the whole trace, whereas a cache that serves the entire campus would achieve an ideal hit ratio of 71%. There are APs with higher ideal hit ratios; for example, an AP in an auditorium had an ideal hit ratio of 73% that corresponds to the 40,064 requests made by six distinct users. These ratios are *ideal* hit ratios, since an infinite size of the cache and cacheability of all shared documents were assumed.

We found that 8% of all requests refer to objects that have been requested by a nearby client within the last hour. This proportion varies widely; at some locations on the campus, 15% of all requests refer to such objects. Also, a lower number of HTTP requests and fraction of repeated requests are made on weekends than on weekdays [255], and several repeated requests exhibit 24-hour periodicity. Assuming that web objects remain in a client’s web browser cache for the entire trace period, the AP-coresident-client cache would attain an ideal hit ratio of 23%, which is less than the ideal hit ratio for same-client and same-AP caches within two minutes.

4.7.5 Same-building and campus-wide repeated requests

Same-building repeated requests are all the requests for which, at sometime in the past, there was another request for the same URL by a client from an AP in the same building where the first request was made. The percentage of such repeated requests (i.e., hit ratio) varies from 15% to 75%.

We investigated how the number of HTTP requests and client population of a building may affect this hit ratio. For each building, the total number of distinct clients that have sent at least one request from an AP in that building

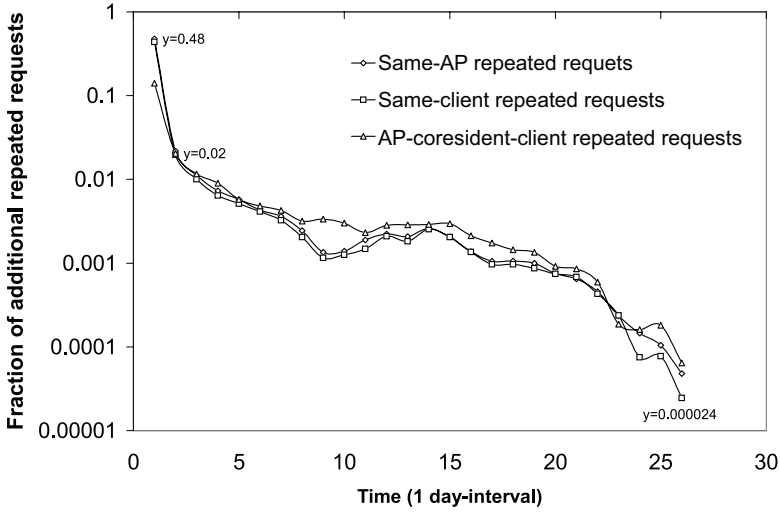


Fig. 4.14. Fraction of additional repeated requests within the entire trace. The number of requests considered is at least 7.6 million. Over 2,800 clients are represented.

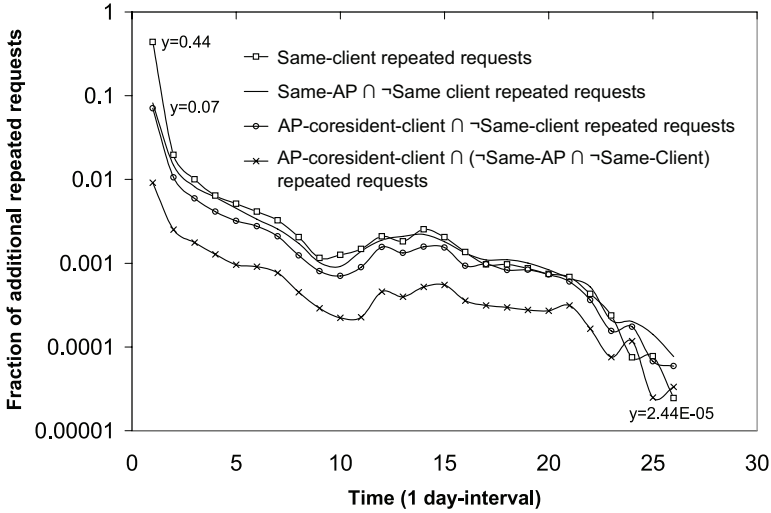


Fig. 4.15. Fraction of additional repeated requests within the entire trace. Impact of each caching paradigm on requests that had a miss on other levels of the cache hierarchy.

represents its *client population* and the total number of requests sent from an AP in that building the *request demand*. The client population varies from one to 1,172 clients, and the request demand ranges from five to 1,929,399 requests. The buildings were sorted in decreasing order with respect to their client population and request demand. In both cases, there is a trend of declining hit ratio. However, the hit ratios across the buildings exhibit high variance and we cannot draw any strong conclusion. It is part of future work to investigate possible correlations of the hit ratios with the building, session, and application type.

4.8 Discussion

The estimation of the wireless workload of an AP has been the epicenter of several measurement-based studies in wireless networks [338, 75, 74, 177, 261, 224, 288, 289, 282, 181, 180]. Most of them present high-level, usually aggregate statistics of the traffic load of APs in campus- or conference-wide networks, or small-scale controlled environments [338, 75, 74, 233, 177, 261, 224, 288, 289, 282, 181, 180]. Temporal and spatial variations in the traffic demand across APs have been reported in several measurement studies on various wireless infrastructures, such as campus WLANs [233, 177, 181], enterprise WLANs [75], and conference hotspots [74, 204]. For instance, in [233], Kotz and Essien characterized Dartmouth's wireless network, examining aggregate traffic and AP utilization. Extending this work, Kotz *et al.* [177] studied the evolution of the wireless network at Dartmouth College using SYSLOG, SNMP, and TCPDUMP traces. They reported the average number of active cards per active AP per day (2-3 in 2001, and 6-7 in 2003/2004) and average daily traffic per AP by category (2-3 times higher in 2003/2004; two or three times more inbound than outbound traffic). Jarosh *et al.* examined issues of congested IEEE802.11b APs in an IETF meeting and made several interesting observations [204]. Measurement-based studies have indicated that several hotspot APs in campus-wide environments exhibit diurnal and weekly periodicities in their traffic load [177, 75, 287, 282].

The application-based characterization of traffic has triggered several research efforts, most of them employing port-based criteria [338, 74, 177]. However, as shown in [268, 215], the majority of emerging applications use random port numbers complicating further the classification problem. Tutschku [341] examined the difference of the uploading from the downloading traffic of a popular peer-to-peer application in a wired network and reported a significant amount of uploaded peer-to-peer traffic. Such asymmetries appeared also in BitTorrent traffic and were highly affected by high-speed downloading [169]. A characterization of online games in terms of user sessions and periodicities of the workload can be found in [109].

Web browsing and peer-to-peer applications dominate the traffic mix in campus-wide wireless infrastructures, accounting approximately for 81% of the

total traffic at UNC. These applications also dominate the traffic mix of most clients. Network management and scanning activity are responsible for 17% of the total flows in our trace. While building-aggregated traffic application usage patterns appear similar, the application cross-section varies within APs of the same building.

Our analysis of the workload of APs in large campus-wide networks revealed interesting structure due to heavy uploading behavior, pervasive log-normality in the system-wide load, and surprisingly heavy distributions of total client associations and roaming operations.

The temporal and spatial locality phenomena of wireless information access and the impact of caching in a large-scale wireless infrastructure were examined in this measurement-driven study. Each client frequently requests objects that it has requested within the past hour, and occasionally requests objects that had been requested by other nearby users within the past hour. The overall ideal hit ratios of user cache, cache attached to an AP, and peer-to-peer caching (where peers are coresident within an AP) paradigms are 51%, 55%, and 23%, respectively. A cache at each AP would achieve an ideal hit ratio of 55% for the entire trace. In general, same-AP caching is beneficial for APs with high hit ratios; such APs were found in the UNC wireless infrastructure. On the other hand, a cache that serves the entire campus would achieve an ideal hit ratio equal to 71%. For a similar user population size in the wired infrastructure of a university campus, the UW study [357] reported a 59% ideal hit ratio. As in the case of wired networks, the single-client locality is a primary factor in wireless data. Thus, there is an opportunity to improve wireless access by more actively caching data in a user cache. Unlike previous studies on wired networks, in which 25% to 40% of documents draw 70% of web access [91], our traces indicate that 13% of unique URLs draw the same number of web access. It would be interesting to examine the spatio-temporal phenomena per application type and wireless environment (such as home, metropolitan, institute, conference, vehicle).

The peer-to-peer caching systems that motivated this study require the objects to be cacheable. Stale objects should not be distributed, but many popular objects on the web are not cacheable by the HTTP standard [132]. It appears that content providers use cacheability to force reloads of their pages for reasons *other* than document freshness (such as the distribution of new advertisements). Although this use of the cacheability mechanisms works well enough in fully connected environments, it is a limiting factor for weakly-connected systems, as the ones described here. Ideally, an object should be cached only for its *true* useful lifetime, while content providers receive the feedback they need.

One of our earlier studies focused on large-scale passive measurements of the characteristics of TCP connections, in terms of their volumes, delays, losses, and lack of termination [180]. Our main findings are summarized as follows:

- The wireless network introduced substantially higher delay variability, but its loss rates were only marginally above those observed for the wired LAN.
- Unnecessary retransmissions are significantly more frequent for wireless clients.
- The number of connections for which the wireless client did not take any action to terminate the connection is significantly larger than the corresponding number of connections of wired clients. The number of interrupted connections are higher for the wireless LAN than for the wired LAN.

Empirical and performance analysis studies indicate dramatically low performance of real-time constrained applications over wireless LANs (such as [62] on VoIP), and large handoff delays [310, 264]. Moreover, mobile users still experience frequent loss of connectivity and high end-to-end delays when they access the wireless Internet. For example, handoff between APs and across subnets in wireless LANs can consume from one to multiple seconds, as associations and bindings at various layers need to be re-established. Unfortunately, such long delays cause disruptions in real-time and streaming applications, such as VoIP and video-on-demand. Examples of sources of delay include acquiring new IP addresses, with duplicate address detection, re-establishing security associations and discovering possible APs without scanning the whole frequency range. The probing operation in the handoff process of the IEEE802.11 MAC is the primary contributor to the overall handoff latency and can affect the quality of service for many applications [264]. As mentioned earlier, the overhead of scanning for nearby APs is routinely over 250 ms, far longer than what can be tolerated by highly interactive applications such as voice telephony.

As popular applications and services from wired networks shift to the wireless arena, new applications emerge, and the use of wireless-enabled devices evolves rapidly, it would be interesting to perform comparative analysis of traces collected from various networking environments. It is important to understand which are the network performance characteristics that have the most dominant impact on the performance of certain applications. Network benchmarks, such as jitter, latency, and packet loss, have been used to quantify network performance. However, what is their impact on how a user “perceives” the performance of its applications? Shifting our attention from MAC- and network-based metrics to application-based characteristics, we plan to address the following issues:

- distinguish the metrics that indicate “*extreme*” *network conditions* (i.e., conditions that degrade substantially the performance of applications)
- quantify user satisfaction and application requirements with more formal subjective and objective metrics/benchmarks
- evaluate the impact of these extreme network conditions on various application-based benchmarks

- understand how user behaviour changes depending on the network topology and technology characteristics

Understanding not only the user demand but also the performance of applications is critical for improving their quality of service and designing effective monitoring and adaptation mechanisms.

Modeling the wireless user demand

This chapter focuses on modeling the wireless user demand, and particularly, the *client associations* and *user traffic demand* in a wireless campus-wide network. It provides a multi-level modeling of the traffic demand and explores the statistical properties of the flows and sessions.

5.1 Introduction

To support wireless networks with better than best-effort service, the deployment of mechanisms for efficient roaming, resource reservation, admission control, caching and prefetching can be essential. For the design and evaluation of those mechanisms, traffic and mobility models in different spatio-temporal scales are required. For example, it would be important to understand the client association patterns and flow demand at different APs. How do clients arrive in a wireless infrastructure? What is the duration of their continuous wireless access and for how long do they stay connected? How do they roam across APs? How do their association patterns differ with respect to device usage pattern, location, and mobility? Which abstractions can be used to model the traffic demand? The above questions drive this research.

It is common practice for a preliminary evaluation of a technology to explore its behavior under well-understood conditions and simple models. Most of the performance analysis studies on wireless network protocols and mechanisms employ traffic models to simulate saturation conditions (asymptotic behavior), e.g., [237, 347, 371, 113, 263, 292, 114, 222, 84, 190, 267, 58, 112, 219, 93]. Other studies simulate UDP flows with fixed packet rate and a few source and destination pairs (e.g., [271, 274, 117, 358, 369]). There are only very few previous studies that employ stochastic packet rate models, such as the Uniform, Poisson, Pareto, Autoregressive (AR), and Markov (e.g., [229, 68, 206]). Example of studies using TCP flows are described in [118, 155, 90, 68, 117], while [210] presents a study which “replays” real-life traces.

Currently, most of the simulators use quite simplistic models, as mobility, topology, access, and traffic models are rich sub-fields on their own, and until recently data from large-scale wireless networks was not available. Typical models used in simulation studies on wireless networks are the following:

- constant bit-rate (CBR) models for traffic
- uniform distribution of clients in an area
- fixed or uniform distribution in the selection of sender and receiver pairs
- fixed arrival of clients at APs
- random-walk based mobility models

It is clear that in several cases the above models are unrealistic. For more comprehensive performance analysis, it is necessary to use realistic and sophisticated models for the parameters of that technology. In general, models should have the following properties:

- accuracy
- robustness
- scalability
- parsimony
- reusability
- “easy” interpretation

Rich sets of empirical traces, collected from large-scale wireless infrastructures, impel modeling efforts to produce more realistic models, and thus, enable more meaningful performance analysis studies. We distinguish the following important dimensions in wireless network modeling, namely, user demand, mobility and access patterns, network topology, and channel conditions. Depending on the environment, the device mobility could be group or individual, spontaneous or controlled, pedestrian or vehicular, known *a priori* or dynamic. In general, network conditions can be characterized by link quality criteria (e.g., packet losses, delays, signal-to-noise ratio), the spatio-temporal distributions of traffic demand and application mix, and the distributions of regions of weak connectivity or no signal (“deadspots”). Network topologies can be described based on their connectivity and link characteristics, distribution and density of peers, degree of clustering, co-residency time, inter-contact time, duration of disconnection from the Internet, and interaction patterns. Highlighting the ability of empirical-based models to capture the characteristics of the user workload, and providing a flexible framework for using them in performance analysis studies are the driving forces of this research.

Figure 5.1 illustrates an example of a wireless infrastructure and a client (client B) roaming between APs before it gets disconnected. Client B first associates with AP 1, it then associates with AP 2, and before getting disconnected from the Internet, it associates with AP 3. While the client is connected to the Internet via the infrastructure, it produces **flows** by receiving and sending packets. An episode of *continuous* wireless connectivity via one or more

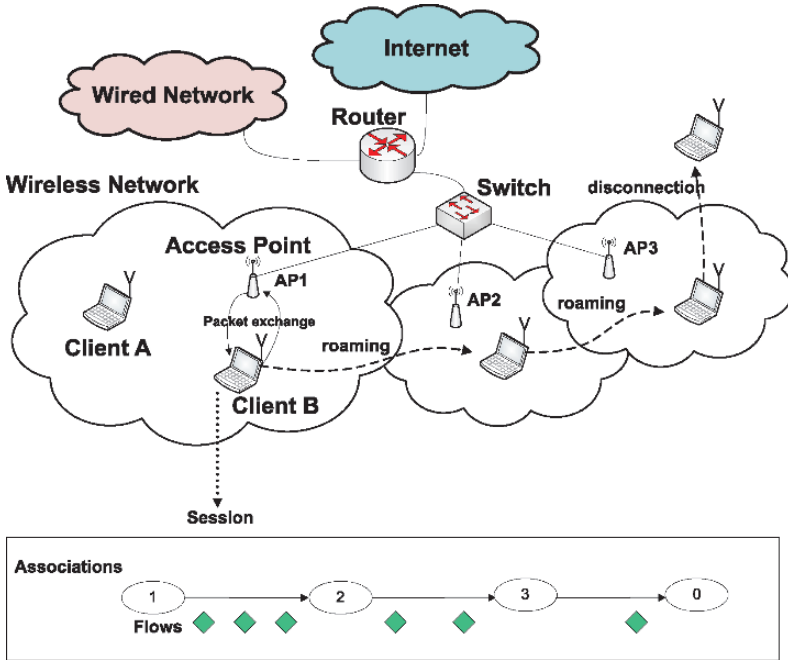


Fig. 5.1. Key components of our traffic models are the sessions and flows.

APs is called a **session**. A session of a client starts when a currently disconnected client successfully associates with an AP and ends when the next disconnection of that client occurs. During a session, a client may visit more than one APs. The wireless life of a client is an alternation between sessions and disconnections. Furthermore, two sessions of the same client cannot be overlapping in time. Sessions are determined by the associations, roaming, and disassociation events of a client. Flows are generated by the various applications running on a device. Sessions and flows are key structures of the user workload analysis, satisfying two objectives:

1. Sessions capture the interaction between the clients and the network infrastructure.
2. Flows are structures with the appropriate level of detail for traffic generation to analyze mechanisms, such as capacity planning, AP selection, and admission control, that motivate this modeling study.

The inherent multi-level spatio-temporal nature of WLANs is intriguing. Our modeling efforts focus on traffic and access demand in large-scale IEEE802.11 networks, aiming to provide a *multi-level* perspective in different

spatio-temporal scales. Table 5.1 shows examples of various spatial and temporal granularities. In fact, selecting the appropriate spatio-temporal scales for

Spatial	AP, client, infrastructure, clusters of APs, building
Temporal	client associations, flow, packet, session, time intervals

Table 5.1. Examples of various spatio-temporal granularities in modeling traffic and access demand.

modeling the characteristics of user workload is an open question that largely depends on the particular mechanism that needs to be analyzed. For example, in the context of capacity planning or admission control, the AP-level can be problematic, since minor changes in the AP infrastructure may impact significantly the workload distribution per AP. Higher levels of spatial aggregation, such as buildings or building types appear to be more appropriate. Similarly, our attention is shifted from the packet-level dynamics and fine time-scales to flow-level modeling. Packet-level dynamics are tightly-dependent on the user mobility, network topology, and channel conditions. Sessions and flows allow us to model user-workload, considering it as a principal building block, independently and complementary to other important dimensions (such as network topology, channel and user mobility).

To evaluate the capability of models to capture the user demand dynamics, we employ various metrics. Although accuracy is an important modeling objective, the scalability and tractability of a model are also critical. This monograph will evaluate the scalability characteristics of the contributed models and addresses the tradeoffs between accuracy and scalability. The models have been extensively validated for *different time periods*, *different spatial and temporal scales*, and periods of *different workload demand*.

The proposed models are based on real-life traces collected at UNC. These traces, the collection process and the generated and correlated flows and sessions are described in [51]. The empirical traces used in modeling the client access and user traffic workload are based on the correlated session and flow data. Section 5.2 discusses the client access patterns, while Section 5.3 focuses on roaming and models the transitions of a client between APs. It presents and evaluates algorithms that predict the next association of a client. Shifting from the client’s perspective to that of the AP, Section 5.4 describes a novel methodology for modeling the arrival process of clients at an AP. Section 5.5 outlines the principal aspects of our modeling approach and presents the proposed models. A flexible framework in which synthetic traces based on various models of user workload can be generated is introduced in Section 5.6. Section 5.7 discusses the scalability and reusability aspects of the user workload. The models are evaluated using statistics- and systems-based metrics in Section 5.8. Section 5.9 describes an analysis of the wireless traffic load at APs using Singular Spectrum Analysis and discusses structural properties of

these time-series. Finally, Section 5.10 discusses related research efforts and Section 5.11 summarizes the main contributions of our modeling efforts.

5.2 Client access patterns

A client initially disconnected from the Internet may associate with an AP in its wireless range. During its visit to that AP, this client may generate traffic by sending and/or receiving packets. Later, the client may reassociate with another AP and prolong its wireless Internet access, or disconnect from the wireless infrastructure. A *transition* is marked by two consecutive connections to distinct APs. Various parameters can be used to characterize the mobility or roaming activity of a client, such as

- duration of sessions and visits
- transitions between APs
- number of inter-building transitions
- duration of time spent and frequency of visits at a certain AP
- duration of disconnection
- predictability of the next AP associations
- arrival process at an AP

Statistics	Visits	Inter-AP transitions	Inter-building transitions
Mean	363	164	32
Median	40	6	0

Table 5.2. Statistics indicating the degree of mobility of wireless clients at UNC considering our trace.

To understand the client access patterns, we analyzed the SYSLOG messages collected at UNC.¹ Wireless clients at UNC exhibit relatively low mobility. On a day, there are 6.8% roaming, 3.7% drop-in, and 2% mobile clients on average. As shown in Table 5.2, the mean of all clients has only 32 inter-building transitions, while the corresponding median is 0. If the average client visits an AP, this AP will be different than the one it is currently connected to, for 48.3% of the time, and it will be in a different building for 13% of the time. In the case of a visit to a different AP, the likelihood that this AP is in a different building is equal to 20.2%.

The locality of the roaming behavior of a client can be also characterized based on the existence of an AP, where that client spends most of its wireless time. To analyze the locality of roaming, the *duration-based homeAP*

¹ These messages were generated from 232 APs between 12:00am on February 10th, 2003 and 11:59pm on April 27th, 2003.

of a client was defined to be the AP (if any) at which this client spends at least a given percentage of its wireless access time. Similarly the *number-of-visits-based homeAP* of a client is the AP (if any) that this client visits most frequently. The threshold for the percentage of wireless access time and the number of visits may vary from 25% to 90%. The duration-based definition

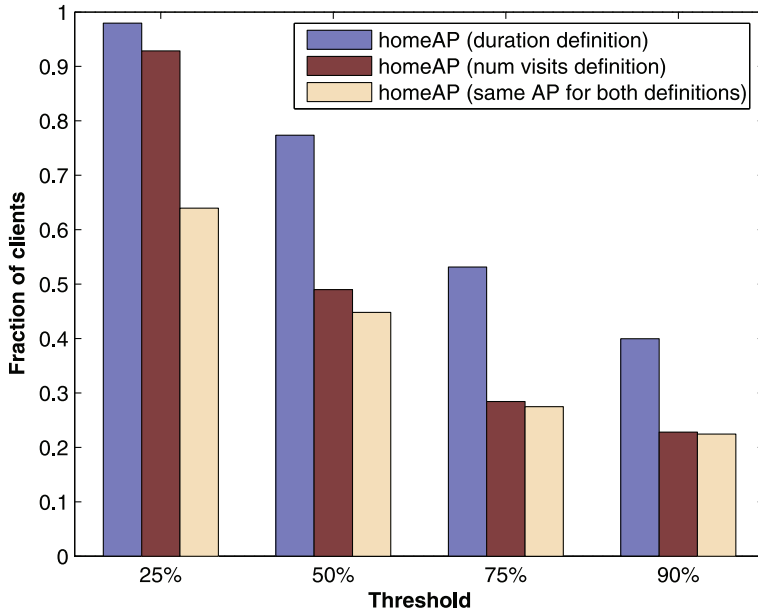


Fig. 5.2. Fraction of clients that have a homeAP for different thresholds according to the two definitions.

is more relaxed than the frequency-based one. More than 50% of the clients spend more than 75% of their time at a single AP, whereas 30% of them visit the same AP more than 75% of the time (as shown in Figure 5.2).

To characterize the roaming of a wireless client, we also defined the *AP path* of a client to be the sequence of *continuous* inter-AP transitions of that client. Similarly the *building path* of a client was defined as the sequence of *continuous* transitions between APs that are located in *different buildings* of that client. A client may potentially have more than one AP and building paths. For example, if a wireless client that was originally disconnected, connects to APs 1, 2, 1, 1, and 10, before disconnecting, its AP path is “1 2 1 10”. The length of this AP path is three. If we assume that the AP 1 and AP 2 are placed in the same building (e.g., building “A”), which is different from the one in

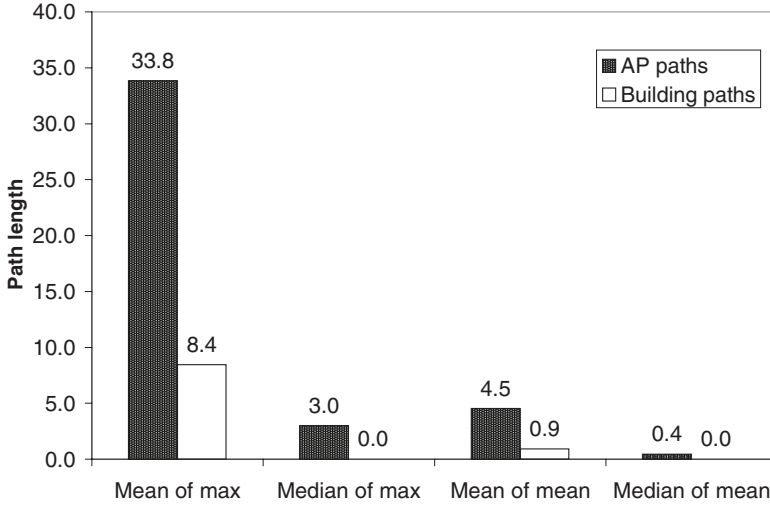


Fig. 5.3. Statistics for the path length of all active clients.

which AP 10 (building “B”) is located, then the corresponding building path is “A B”. Figure 5.3 shows the mean and median for the maximum and mean AP path, and building path length of all users.

Sessions are categorized according to client mobility and are classified as stationary and mobile. Stationary sessions are composed of associations at APs located in the same building. Mobile sessions can be further divided into those with a transition between two buildings (“one-edge”) and all the others with transitions to several pairs of buildings (“multiple-edge”). As the number of edges increases, the mobility of a client is considered to increase.

5.2.1 Session duration

A session reflects an episode of continuous wireless access of a client at an infrastructure. During a session, the infrastructure needs to be capable of supporting its clients, and thus, we were interested in understanding the session duration. We found that 56.4% of the sessions lasted less than 30 minutes, 68.9% less than one hour, and only 16.2% less than one minute. The vast majority of the stationary sessions last 1.5 hours or less while the medians of stationary, one-edge and multiple-edge session duration are 9, 18, and 34 minutes, respectively. To compare the duration of different types of sessions, we employed the concept of *stochastic order*. A random variable X is *stochastically larger* than another random variable Y if

- $P(X > t) \geq P(Y > t)$ for every t , and

- $P(X > t) > P(Y > t)$ for some t .

To compare the duration of mobile and stationary sessions, the notion of stochastic order between two distributions is used [105]. As becomes apparent from their complementary cumulative distribution function (CCDF), the duration of mobile sessions is *stochastically larger* than the duration of stationary sessions. As the session mobility increases, the session duration also increases stochastically.

The CCDF of the stationary session duration has two nearly linear regimes. This led us to propose to model the stationary session duration using a *Bi-Pareto* distribution.² A BiPareto distribution's CCDF is given by

$$\left(\frac{x}{k}\right)^{-\alpha} \left(\frac{x/k+c}{1+c}\right)^{\alpha-\beta}, \quad x \geq k.$$

A BiPareto distribution has four parameters (α, β, c, k) , that can be estimated via maximum likelihood. The scale parameter k ($k > 0$) is the minimum value of the BiPareto random variable. The CCDF initially decays as a power law with exponent $\alpha > 0$. Then, in the vicinity of the breakpoint kc (with $c > 0$), the decay exponent gradually changes to $\beta > 0$. Notice that on log-log plots, a CCDF of the form $x^{-\alpha}$ would appear as a straight line with slope $-\alpha$. A BiPareto distribution with $c = 0$ corresponds to a Pareto with parameters (β, k) .

The BiPareto distribution was fitted to the stationary session duration, and the parameters were estimated to be (0.05, 0.76, 867.64, 1) using the maximum likelihood method. Figure 5.4 compares the empirical log-log CCDF with the theoretical log-log CCDF of the fitted BiPareto distribution (the two linear regions also indicated). The two CCDFs closely follow each other with a coefficient of determination of 0.99. The major difference appears in the tails, which only concerns 1% of the sessions. One possible explanation for this discrepancy is due to censoring caused by our data collection period. Because of this, we did not observe any stationary sessions that are longer than the collection period. Otherwise, those long session durations may bring the tail closer to the BiPareto tail.

Several other common parametric distributions were also examined, such as the Lognormal, Weibull, and Gamma (see Appendix) but the BiPareto gave the best fit. In fact, the fit became even better by aggregating the durations into minute resolution level, which could be fitted with a BiPareto with parameters (0.34, 1.37, 258.94, 1).

The log-log CCDFs for mobile sessions also exhibit two linear regions up to three hours (Figure 5.5). We truncated the mobile session durations at three hours and modeled them using a truncated BiPareto distribution. The truncation percentage is about 9% and the fitted parameters for the mobile session durations are (0.02, 1.42, 1633.42, 1).

² More details on the BiPareto distribution and its estimation method can be found in [278].

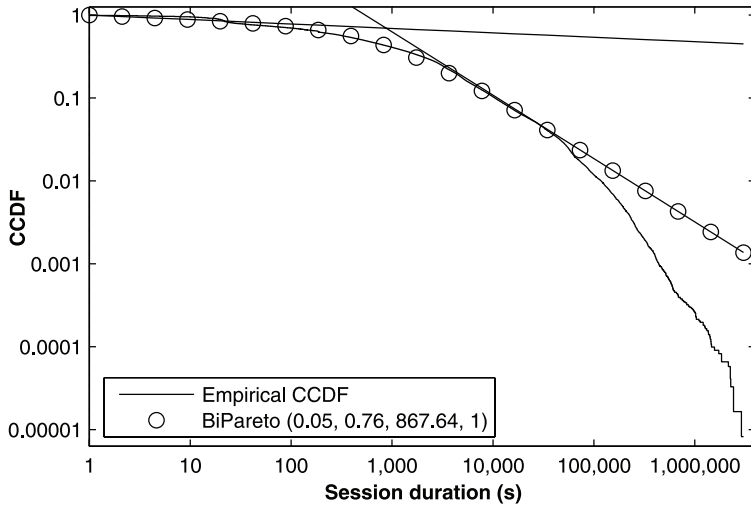


Fig. 5.4. Stationary session duration (empirical and model).

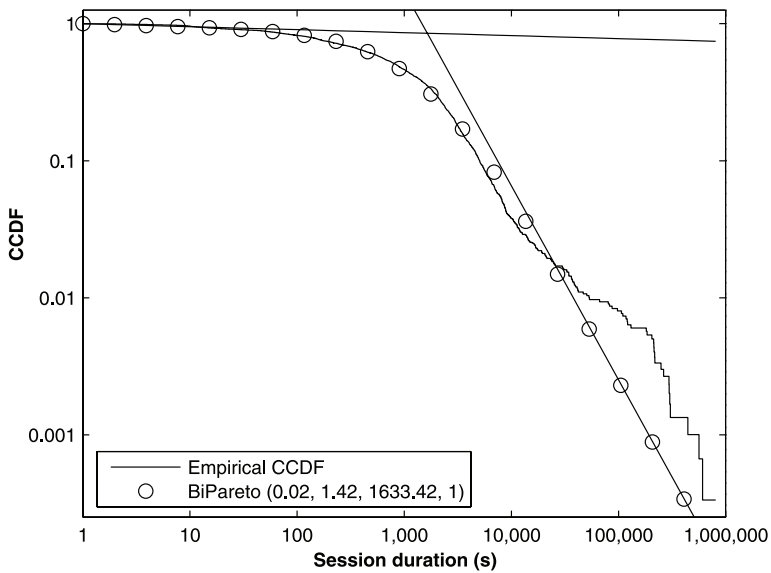


Fig. 5.5. Mobile session duration (empirical and model).

5.2.2 Transient sessions

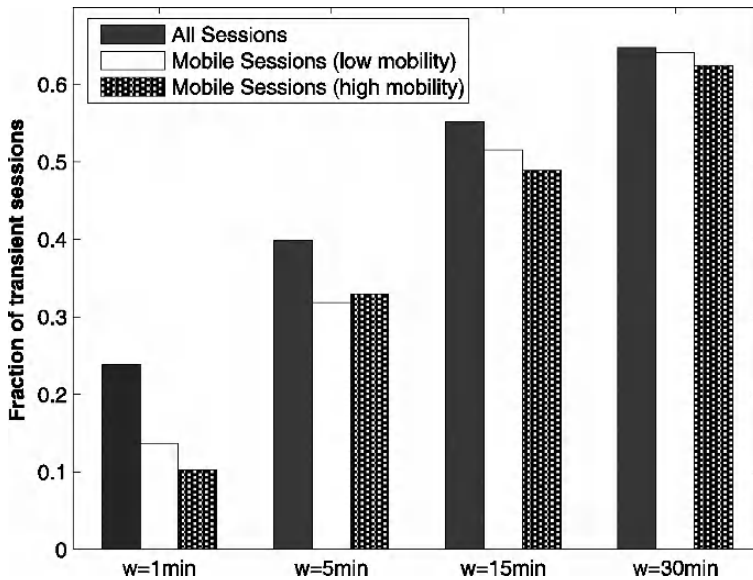


Fig. 5.6. Fraction of transient sessions (i.e., all visits to a building in the building path last less than w).

To further characterize the access patterns, the distribution of the duration of visits within a session was examined: are most of the sessions composed of relatively short visits (at APs)? Are the durations of visits in the same session “statistically similar”? Does the first visit differ statistically from the last? Based on the duration of visits at each *building* involved in a session, a *transient* session is defined as a session that does not have any visits to a building that last more than a certain number of minutes. Figure 5.6 illustrates the distribution of transient sessions for different time periods varying from one to thirty minutes. By increasing the time period, the fraction of transient sessions also increases. However, for low thresholds (e.g., one or five minutes), the mobile sessions tend to be less transient than the rest. More than 20% of the clients have at least 90% of sessions in which all their visits last 30 minutes or less (as shown in Figure 5.7).

To further analyze how the session time is distributed among its visits, the percentage of visits that have a duration close to the median visit duration of that session was computed. Interestingly, 50% of the sessions have less than 10% of their visits in the 10% interval of the median duration of their session.

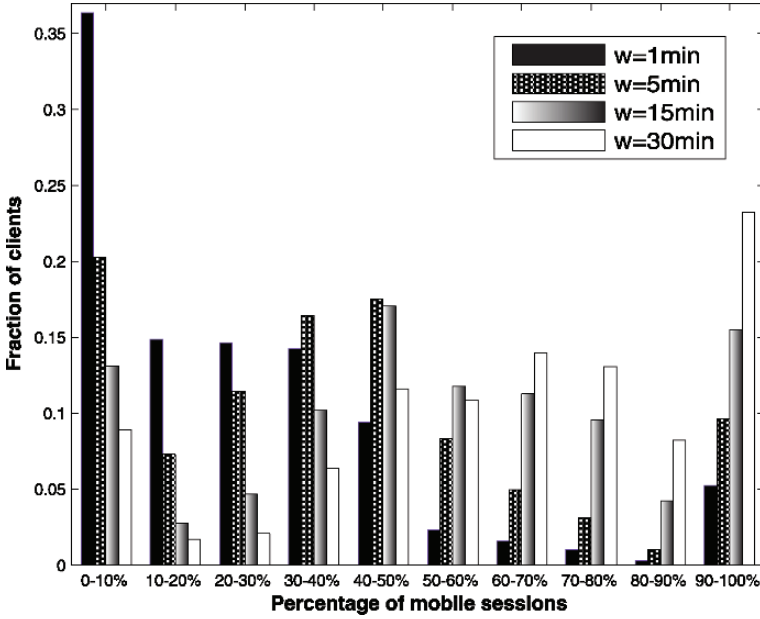


Fig. 5.7. Fraction of clients that have a certain fraction of their sessions transient.

Mobile sessions tend to have a small percentage of long visits and a large percentage of short visits. As a result, sessions with high mobility are less transient, as it is less likely for all visits to fall below a certain threshold. This indicates that all our results are consistent with each other. Figure 5.8 does not include the stationary sessions, since by definition, they have only one visit and their similarity index is 100%. “Mobile Sessions” is a subset of “All Sessions”, including only these sessions with visits to two or more APs located at different buildings.

5.2.3 Revisits

Wireless users may revisit the APs of an infrastructure multiple times. Caching data at an AP can mask delays that roaming users experience during an association process. To get an insight into how long a user’s data (e.g., profile, cache) should be stored in an AP, we estimated how likely it is for that client to revisit an AP within a certain time interval.

For each client, we used its state history with a timestamp indicating when the client visited each state. Its probability of revisiting a state (i.e., revisit probability of this client) in a given time interval is defined as the fraction

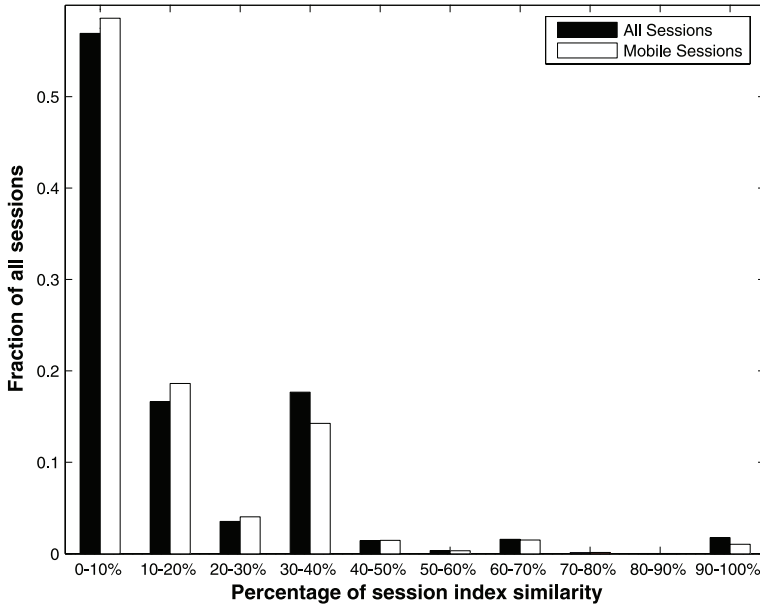
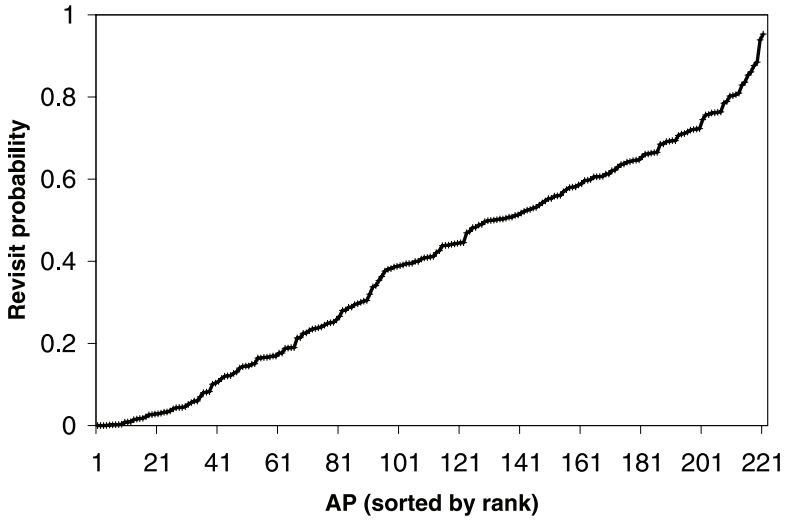


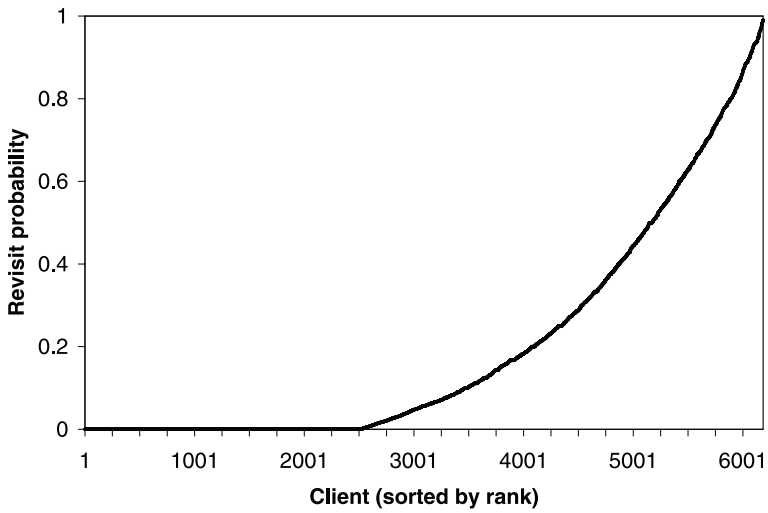
Fig. 5.8. Percentage of visits in a session with duration within an interval of $\pm 10\%$ from the median visit of that session. The similarity index of a session was defined as the percentage of visits that are within a certain interval of their median visit duration, such as $\pm 10\%$ from the median duration of the visits in that session.

of times this client visits that state within a time period since its last visit to that state and there is also at least one visit to *any other non-zero* state between these two consecutive visits to that state. Furthermore, the revisit probability for an AP is defined as the fraction of all visits which are revisits at that AP, considering the entire client population.

The mean revisit probability for a one-hour interval is 20% for clients and 40% for APs. The revisit probability varies drastically among APs and clients, varying between 0% and 95% among APs, and 0% and 99% among clients. Figure 5.9 (a) shows the probability for each AP that a visit at that AP was a revisit, while Figure 5.9 (b) shows the probability that a visit was a revisit for each client. The median revisit probability was 40% and 6% for APs and clients, respectively. Therefore, a cache with a lifetime of one hour at each AP would be beneficial.



(a) For each AP



(b) For each client

Fig. 5.9. Revisit probability.

5.3 Roaming across APs

While a client roams in a wireless infrastructure, it may associate with multiple APs. During an association process, a client requests access from an AP and that AP accepts or declines the access. The association overhead in addition to end-to-end delays could be prohibitive for several real-time applications running on roaming clients. By profiling a client and predicting its next associations, these delays could be masked. Specifically, a roaming controller could keep track of a client and its application access. It could predict the next association of a client and prefetch data on its behalf. Furthermore, APs can use the next-associations of clients and traffic demand to perform load balancing and better utilization of their buffer and wireless bandwidth. The association protocol could be also enhanced by advising a client to avoid extremely busy APs. These issues motivate our next-AP prediction algorithms.

Based on a client's state history, an algorithm that predicts the n -th state of the client can be developed. Our prediction is based on a Markov-chain model and uses the current state to predict the next one. For each client, a first-order Markov-chain based on the client's state history is generated. Each state of the Markov chain corresponds to a *state* as defined in Section 4.4. Let us denote the set of all the states as S . The transition probability from state j to state k is the relative frequency of the sequence of states $s_j s_k$ in the client's state history ($s_j, s_k \in S$). This corresponds to the (j, k) entry of the transitional probability matrix $P^{(1)}$. The prediction model can be extended by using the previous as well as the current state to predict the next one.³ In this prediction model, we computed the relative frequencies of $s_i s_j s_k$ ($s_i, s_j, s_k \in S$). This corresponds to the (i, j, k) entry in the three-dimensional matrix $P^{(2)}$. We evaluated the following variations of such prediction algorithms using different amounts of history:

One-state history: This model is the one-state history as discussed above. The first $n-1$ states are used to build the matrix $P^{(1)}$. Given that the current state is s_j , we predict the next state to be the state:

$$\arg \max_{s_k} \{P^{(1)}(j, k), \forall s_k \in S\}. \quad (5.1)$$

The error in making this single prediction of the next state n is $\varepsilon_n = 1 - P^{(1)}(j, k)$.

One-state window: If the $(n-1)$ -th state occurs at time t , this model uses the sequence of states that occur between $t-24$ hours and t to build its probability matrix. Then, it predicts the n -th state in the same way as the one-state history model.

³ There are some storage considerations. For example, a very mobile client can visit half of the total number of APs. Storing a single client's three-dimensional matrix M for 128 APs, for a single day, requires about 8 MB of memory, while a four dimensional matrix would require about 1 GB of memory.

Max of one-state window and history: This model compares the probability of the one-state window and history algorithms and reports the state that is more probable.

Two-state history: This model is the two-state history model discussed above. The first $n - 1$ states are used to build $P^{(2)}$. Let the $(n - 2)$ -th state be s_i and the $(n - 1)$ -th state be s_j . The next state is predicted to be the state s_k such that k maximizes $P^{(2)}(i, j, k)$. The error in the single prediction of the next state n is $\varepsilon_n = 1 - P^{(2)}(i, j, k)$.

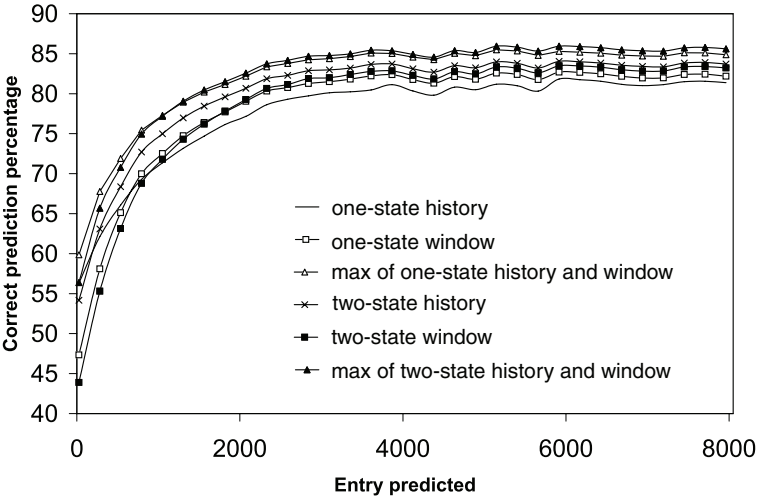
Two-state window: If the $(n - 1)$ -th state occurs at time t , this model uses the sequence of states that occur between $t - 24$ hours and t . It then predicts the n -th state in the same way as the two-state history model predicts the next state.

Max of two-state window and history: This model compares the probability of the two-state window and history algorithms and reports the state that is more probable.

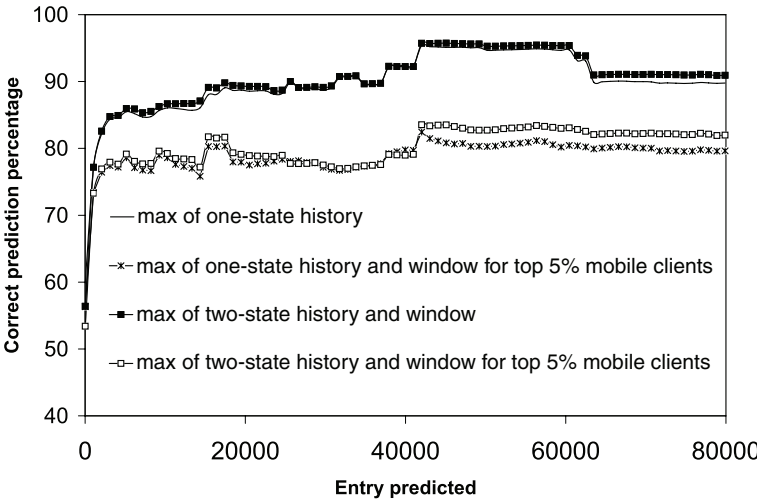
To evaluate the performance of these prediction algorithms, we employed the following metrics:

- the *correct prediction percentage*, which is the percentage of times that the next state was predicted correctly
- the *prediction error after predicting n states*, which is defined as the mean of the error of all predictions made

The training set of each client consists of its first 25 SYSLOG entries. The mean correct prediction percentages for predicting state 8,012 were 81.36%, 82.16%, and 84.85% for the one-state history, one-state window, and max of one-state window and history models, respectively. For the two-state history, two-state window, and max of both two-state window and history model, the mean correct prediction percentages were 83.68%, 83.19%, and 85.59%, respectively. Figure 5.10 (a) illustrates the percentage of correct predictions after each entry. The one-state history and the one-state window history model have similar performance, while the two-state models perform slightly better than their one-state counterparts. The one-state history, one-state window, and max of one-state history and window had prediction errors of 0.26, 0.23, and 0.21, respectively, and their two-state counterparts had prediction errors of 0.23, 0.22, and 0.20, respectively. The standard deviations for the correct prediction percentages are less than 0.19 for the one-state models and less than 0.18 for the two-state models. The standard deviations for the prediction error are less than 0.23 for the one-state models and less than 0.21 for the two-state models. Note that by maintaining information about the last 2,000 entries, the max of two-state history and window achieves a correct prediction percentage of at least 82.17%. This suggests that if storage space is a concern, the model can be implemented in a slightly different manner that uses only a certain number of entries, so that it is space efficient and still has a high correct prediction percentage.

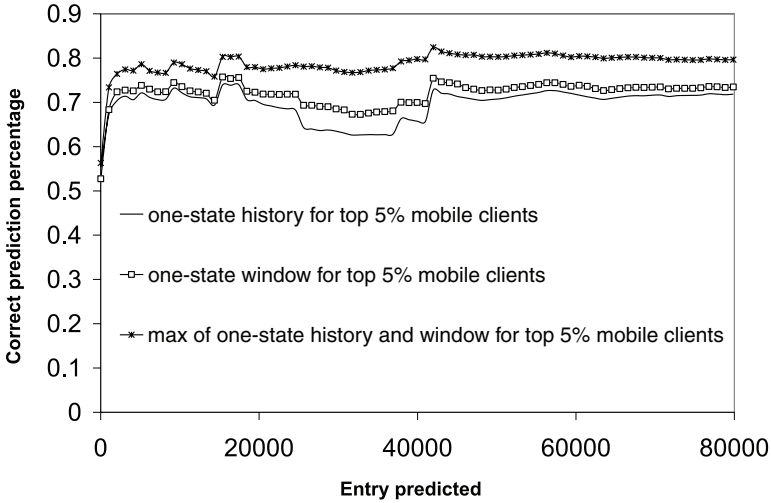


(a) All clients (up to 8,000 entries)

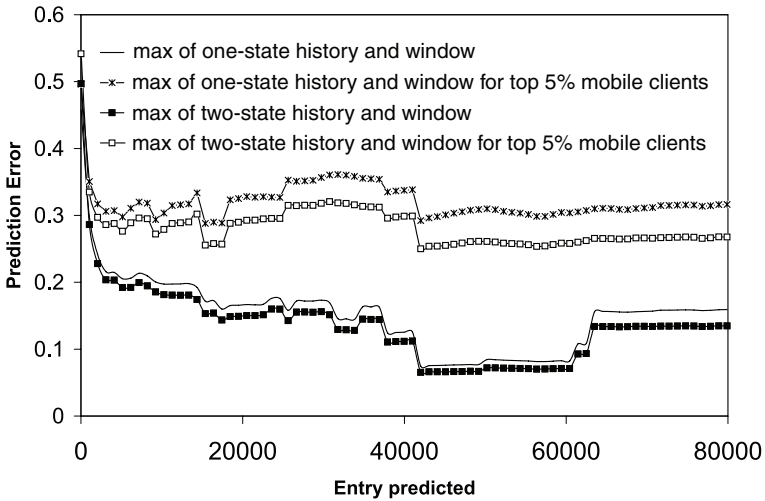


(b) Comparing all vs. only mobile clients (up to 80,000 entries)

Fig. 5.10. Next-state prediction algorithms. Correct prediction percentage after each entry. Initially, there were over 3,900 clients. Only 31 clients participated for the prediction of the last entry.



(a) Only top 5% mobile users



(b) All clients

Fig. 5.11. Next-state prediction. Prediction error after each entry. Initially, there were over 3,900 total clients and 300 mobile clients. Only two clients in total—one mobile client—participated for the prediction of the last entry.

The top five percent of clients—in terms of total number of inter-building transitions, which also have 8,012 events—have a correct prediction percentage of 79% for predicting state 8,012. Figure 5.10 (b) illustrates the correct prediction percentage after 80,000 instead of 8,000 entries and Figure 5.11 shows the corresponding prediction errors. Figure 5.11 (a) focuses on the top five percent of clients that exhibit the highest degree of mobility (considering their number of inter-building transitions). The max of one-state history and window algorithm performs reasonably well, is simple, and does not have high memory requirements.

We also incorporated a time component into the sequence of states, as described in [83]. This method produces additional states by polling for a client’s state at regular time intervals and thereby creates a state history based on both movement and time. For clients disconnected for long time intervals, this polling process introduces long sequences of 0, resulting in an overestimation of the performance of the predictor. Thus, this movement and time model can be used to predict the next state only for connected clients. The mean correct prediction percentage, using the max of two-state window and history model was 87% at the last entry, for which there were more than 30 clients.

5.4 Arrivals of wireless clients at APs

The client associations can be studied either from the perspective of the client, as transitions between APs, or from the perspective of an AP, as arrival processes. Section 5.3 presented a Markov-based model for the client transitions to APs. Here, we describe a novel methodology for modeling the arrival process of clients at APs as a *time-varying Poisson process* with different arrival rates. Poisson models have been already used to characterize the “arrivals” of humans to the Internet, i.e., the times at which humans access the Internet to preform a task conform to a memory-less process with an arrival rate that can be constant over time intervals of many minutes to perhaps an hour.

A quantile plot with simulation envelope is used for testing the goodness-of-fit. Furthermore, we investigated the impact of the type of building (i.e., its functionality) in which the AP is located at the arrival rate and cluster these visit arrival models based on the building type. In addition to each AP’s unique IP address, we maintained information about the building the AP is located in, its *type*, and its coordinates. The possible building types in our study are the following:

- academic
- administrative
- athletic
- business
- clinical

- dining
- library
- residential and letter society
- student stores
- health affairs
- playing fields, performing arts, and theater

This study focused on the visit arrivals at the sixteen hotspot APs of the UNC wireless infrastructure. An AP is defined as *hotspot* when it belongs in the intersection of two sets, namely the top 5% APs based on the total maximum traffic and the top 5% APs based on the maximum hourly traffic. The distribution of the selected hotspot APs per building type is as follows: academic (8 APs), library (3 APs), residential (3 APs), and theater (2 APs).⁴

5.4.1 Time-varying Poisson process

In this section, the focus shifts from a client's perspective to an AP's and explores the wireless access by modeling the arrivals of clients at an AP. Specifically, this section models the arrivals of clients at an AP as a *time-varying Poisson process*. For this purpose, let us first introduce the definition of a time-varying Poisson process and then construct a test for such a process.

Let $\{\Lambda(t) : t \geq 0\}$ be a stochastic point process, which counts the number of events (or arrivals) in the interval $[0, t]$. Sometimes, $\{\Lambda(t)\}$ is referred to as the arrival process of the events of interest. For example, let $\{\Lambda(t)\}$ be the arrival process of client visits at a particular AP. $\{\Lambda(t)\}$ is a Poisson process if it satisfies the following two properties:

1. The number of arrivals in disjoint intervals are independent
2. For some finite $\lambda > 0$, $P(\Lambda(t) = j) = e^{-\lambda t}(\lambda t)^j / j!$, $j = 0, 1, \dots$

Thus, for each t , $\Lambda(t)$ is a Poisson random variable with mean λt , the product of the arrival rate λ and the interval length t . Note that a Poisson process is a renewal process, where the inter-arrival times are independent exponential random variables [320]. It is well-known that such a process results from the following behavior: there exist many potential, statistically identical arrivals; there is a very small—yet non-negligible—probability for each of them arriving at any given time; and arrivals happen independently of each other.

One closely related variation is a time-varying Poisson process, where the arrival rate is a function of time t , say, $\lambda(t)$. Such a process is the result of time-varying probabilities of event arrivals, and it is completely characterized by its arrival rate function. A smooth variation of $\lambda(t)$ is usual in both theory

⁴ In our analysis, the hotspot APs with unknown building type and location were excluded. We had only limited information about the exact functionality of the rooms in which the hotspots were located. For example, APs in academic buildings could be found in classrooms, offices for advising students, lounges, and meeting rooms. Hotspot APs in residential buildings were found in lounges and labs.

and practice in a wide variety of contexts, and seems reasonable for modeling client visits to an AP.

Construction of a statistical test

We constructed a test for the null hypothesis that an arrival process is a time-varying Poisson process, with a slowly varying arrival rate. In statistics, a null hypothesis is a hypothesis set up to be nullified or refuted in order to support an alternative hypothesis. When it is used, the null hypothesis is presumed true until statistical evidence—in the form of a hypothesis test—indicates otherwise [33].

To begin with, we broke up the interval of a day into relatively short blocks of time. For convenience, blocks of equal length, L , were used, resulting in a total of I blocks; however, this equality assumption can be relaxed. For the analysis in Section 5.4.2, L was set to be six minutes.

Let T_{ij} denote the j th ordered arrival time in the i th block, $i = 1, \dots, I$. Thus $T_{i1} \leq \dots \leq T_{iJ(i)}$, where $J(i)$ denotes the total number of arrivals in the i th block. Define the variables $T_{i0} = 0$ and, for $j = 1, \dots, J(i)$,

$$R_{ij} = (J(i) + 1 - j) \left[-\log \left(\frac{L - T_{ij}}{L - T_{i,j-1}} \right) \right]. \quad (5.2)$$

Under the null hypothesis that the arrival rate is constant within each time interval, the $\{R_{ij}\}$ will be independent standard exponential random variables as is proved below.

Let U_{ij} denote the j th (unordered) arrival time in the i th block. Then, the assumption of a constant Poisson arrival rate within this block implies that, conditioned on $J(i)$, the unordered arrival times are independent and uniformly distributed between 0 and L . If we define $V_{ij} = \frac{L}{L - U_{ij}}$, then, it follows that V_{ij} are independent standard exponential random variables. Indeed, notice that $T_{ij} = U_{i(j)}$, and thus

$$V_{i(j)} = \log \left(\frac{L}{L - U_{i(j)}} \right) = \log \left(\frac{L}{L - T_{ij}} \right).$$

Here, (j) indicates the j -th order-statistic.⁵ Evidently, $R_{ij} = (J(i) + 1 - j) (V_{i(j)} - V_{i(j-1)})$. Then, the exponentiality of R_{ij} results from the following lemma.

Lemma: Suppose that X_1, \dots, X_n are independent standard exponential random variables, then $Y_i = (n - i + 1)[X_{(i)} - X_{(i-1)}]$, $i = 2, \dots, n$, are also independent standard exponential random variables. Any common test for the exponential distribution can then be applied to R_{ij} for testing the null hypothesis. For convenience, the familiar Kolmogorov-Smirnov test[125] was

⁵ For example, consider a sample of n numbers: U_1, \dots, U_n . The term $U_{(j)}$ indicates the j -th smallest one.

used. This nonparametric test is based on the maximum deviation between the empirical CDF of the data and the hypothesized theoretical CDF. The exponentiality can also be tested using a graphical tool, such as an exponential quantile plot.

5.4.2 Arrival process of visits at wireless hotspots

As an illustration, this section now analyzes the arrival process of client visits at the hotspot AP 222, which is located in an academic building. The analysis has been also validated using APs in other building types.

Exploratory data analysis

In an exploratory data analysis, we employed the SIgnificant ZERo crossing of the derivatives (SiZer) map [110], a powerful visualization method which enables statistical inference for discovery of meaningful structures within the data. It identifies important underlying structures, and not artifacts due to sampling noise. SiZer is based on scale-space techniques used in computer vision [253]. Scale-space is a family of locally linear smoothed data curves indexed by the scale, which is the smoothing parameter or the bandwidth h used for the local linear smoothing [145]. The bandwidth controls the level of smoothing, and can be treated approximately as the window size used for computing local averages in order to smooth the data. SiZer considers a wide range of bandwidths to derive “smoothed versions of the underlying curve” at various resolution levels. This approach then visualizes all the information available in the data at each given scale. A detailed introduction to SiZer, along with more examples and software, can be found in [110]. An illustration of the use of SiZer to analyze flow arrival processes is available in [258].

This analysis indicated that the arrival rate appears to be time-varying at all scales. At coarse scales (or large bandwidths), there is an overall daily increasing trend; at medium scales, the rate function first decreases between early morning and 14:00, and starts increasing until 18:00, before decreasing again. More features appear at fine scales, suggesting that the arrival rate has several ups and downs during this period.

In order to better examine these features, we focused on the hour between 17:30 and 18:30, which has the largest arrival rate and consists of 2143 visits. First, we calculated the inter-arrival times between every two consecutive sorted visits. The visual inspection of the corresponding SiZer maps indicates that the inter-arrival times may take only a finite number of discrete values. Furthermore, there is an artifact caused by the rounding of visit arrival times to nearest whole seconds. To compensate for this rounding effect, we “unrounded” the data by adding independent uniform $(-0.5, 0.5)$ noise to each visit’s start time before calculating the inter-arrival times. The distribution of the inter-arrival times is analyzed in Figures 5.12 and 5.13. Note that the

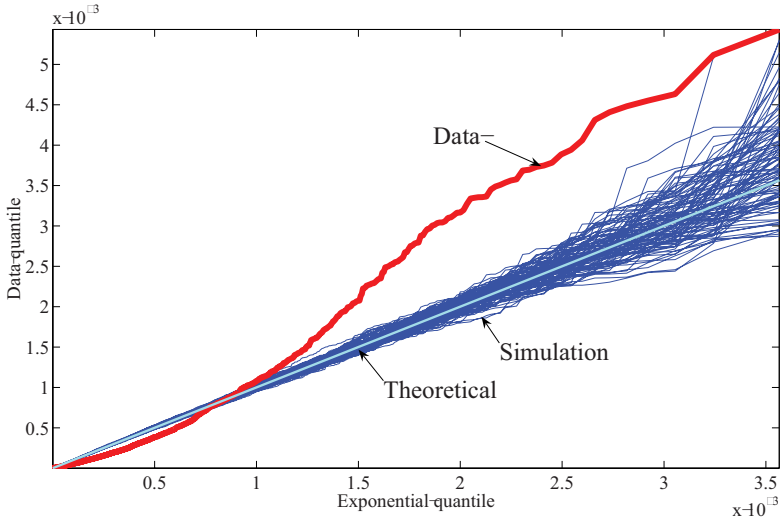
distribution of the inter-arrival times is exponential, if the arrival process is Poisson.

An exponential quantile plot, which can be used as a graphical method for assessing the goodness-of-fit of the exponential distribution on the inter-arrival times, shows that the exponential distribution does not fit well our data (“Data” vs. “Theoretical” in Figure 5.12 (a)). The wider, dark-grey curve (marked as “Data”) is the main quantile plot, which plots the *actual* data quantiles (based on our traces) versus the corresponding *theoretical* exponential quantiles (brighter, thinner curve, marked as “Theoretical”). The parameter for the theoretical distribution is estimated using maximum likelihood. When the theoretical distribution is a good fit, the “Data” curve should closely follow the diagonal “Theoretical” line. To account for possible sampling variability, an envelope of 100 overlaid curves is constructed. Each of these curves is a similar quantile plot, where the corresponding data are simulated from the theoretical exponential distribution. This envelope provides a simple visual accounting for the sampling variability. When the theoretical exponential distribution fits the inter-arrival times well, the “Data” curve should lie mostly within the envelope. The observed substantial departure of the curve from the envelope in Figure 5.12 (a) strongly suggests that the inter-arrival times are not exponentially distributed.

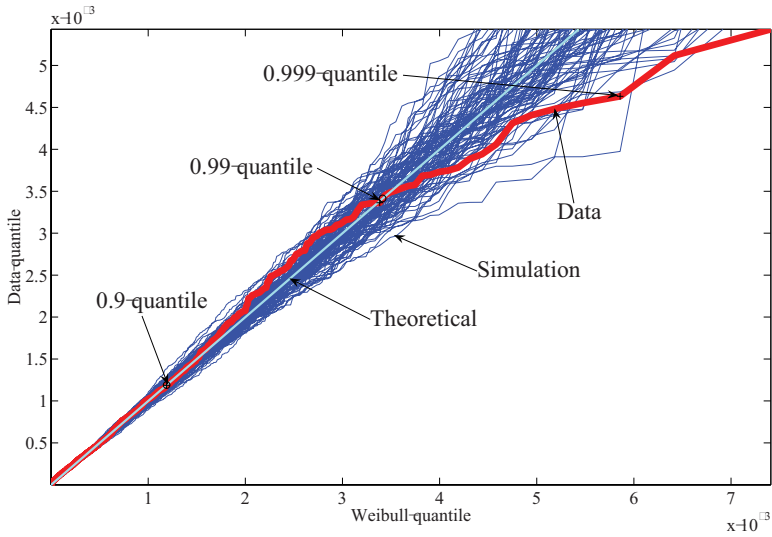
Figure 5.12 (b) shows a Weibull quantile plot for the inter-arrival times. The two parameters of the Weibull distribution are fitted by matching the 90th and 99th percentiles of the data and the theoretical distribution. The plot indicates that the inter-arrival times follow approximately a Weibull distribution. In addition, the inter-arrival times in our data are not independent as shown in the corresponding auto-correlation plot in [290]. All the auto-correlations are significantly positive. The strong auto-correlation of the inter-arrival times suggests that the visit arrival process can not be modeled as a renewal process with independent Weibull inter-arrival times. A more appropriate model is to combine Weibull inter-arrival times with a suitable dependence structure, as proposed in [120]. Generating or simulating such a dependent process is much more complicated, since one has to specify the dependence structure. We decided to model our data using a time-varying Poisson process as it has a nice practical interpretation and is easier to simulate.

Time-varying Poisson process for visit arrivals

We applied the statistical test proposed in Section 5.4.1 and drew an exponential quantile plot to show that the arrival process of client visits at AP222 is a Poisson process with a time-varying arrival rate. The analysis was carried out in detail for the process between 17:30 and 18:30 only. We broke the hour into ten six-minute intervals, and calculated the R_{ij} according to Eq. (5.2) by setting L to be six minutes. The corresponding Kolmogorov-Smirnov test statistic is 0.0188, and has a p-value of 0.15 with 2143 observations, which means that the null hypothesis can not be rejected.



(a) Exponential quantile plot



(b) Weibull quantile plot

Fig. 5.12. Quantile plots for visit inter-arrival times between 17:30 and 18:30.

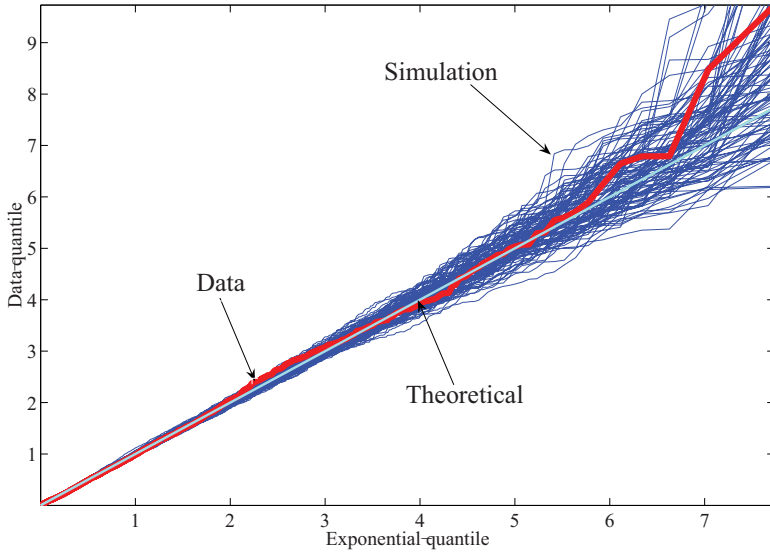


Fig. 5.13. Exponential quantile plot for R_{ij} between 17:30 and 18:30 (AP 222).

Figure 5.13 shows the exponential quantile plot for the R_{ij} , which clearly suggests that they are exponential. The maximum likelihood estimate for the exponential parameter is 1.0024, which is very close to 1, and this agrees with the claim that the R_{ij} are standard exponential random variables. The corresponding auto-correlation plot suggests that the R_{ij} s are approximately independent [290]. Thus, the null hypothesis that the visit arrival process within an hour is a time-varying Poisson process is validated both mathematically and graphically. There are well-developed methods for simulating time-varying Poisson processes, such as the *thinning* method described in [245, 354]. Along with models for visit durations, we can also generate synthetic traces.

An interesting question is whether or not the functionality of an area affects the arrival rate of client visits at APs located in that area. In general, we have limited information about the exact activities, schedules, and usage of the areas around APs, some of them also used for diverse activities. However, we do observe clusters of hotspot APs with similar arrival patterns according to their functionality.

Three clusters of APs can be distinguished: the first cluster contains APs placed in libraries, the second one in lounges of residential buildings, and the third one in meeting rooms and lounges in theaters (recreational centers). For each AP, we also calculated the 25th-percentile, median, and standard deviation. We found a similarity among APs located in meeting rooms and

lounges of theater/recreational building types. APs placed in lounges or labs in dorm/residential areas have similar patterns which differ from the ones in residential buildings. Similarly, APs located in classrooms have similar patterns, which actually differ from the visit arrival pattern in a classroom, where lectures for a middle school take place or the area with offices for advising students. We also observed a similarity in the arrival patterns at the three library APs.

5.5 Methodology for modeling user demand

There is a consensus in the network community that traffic modeling should not address elements that are dominated by too specific network-side characteristics or conditions. Otherwise, simulations and experiments using the respective models can *never* study changes in these conditions or new network mechanisms that shape them. For example, in the context of WLANs, modeling the precise sequence of associations and disassociations inside sessions is network-specific and non-deterministic, since small changes in the network layout, physical environment, radio propagation, and network/client equipment can dramatically change the association/disassociation dynamics. A new proposed algorithm for AP selection may also change association dynamics. Therefore, the simulation model should not impose *a priori* a certain sequence of associations and disassociations. This requirement is satisfied when sessions are the subject of modeling. The simulated session may end up having completely different association dynamics, but the corresponding workload (i.e., traffic generated during a time period) is preserved.

As mentioned in Section 5.1, we distinguished four important dimensions in modeling wireless networks, namely, user traffic workload, user mobility, network topology, and link conditions. In a performance analysis study, one can integrate the proposed user workload models (or corresponding traces) with the appropriate user mobility, network topology, and channel condition models, depending on the specific characteristics of the environment/setting.

5.5.1 Sessions and flows

In our approach, sessions represent the highest-level unit of wireless network traffic load, including all the packets sent and received by the APs due to the client's communication with one or more Internet hosts. Working with flows, such as TCP connections and UDP conversations, is in line with the approach taken in [278, 261, 294] and the principles of network-independent modeling described in [295]. Network flows are well-separated collections of packets between a pair of Internet hosts, sharing the same transport-layer "5-tuple". Simulating the user demand consists of simulating the sessions and flows initiated inside them, without trying to capture packet-level and association dynamics that are closely affected by the underlying network mechanisms.

User demand can be simulated at both the client association and flow levels by using models of the compound process of sessions and flows. As shown here, sessions have a well-behaved arrival process, which can be accurately described using a time-varying Poisson process. As previously discussed, the Poisson process is a parsimonious model that has been used widely to model the arrival process of events initiated by humans (e.g., in a telephone network or in the Internet). The session arrival process provides the seeds of a cluster process, in which the arrivals of sessions imply the arrivals of correlated sets of flows. The following parameters are modeled:

- session arrivals
- number of flows within a session
- flow inter-arrival times within a session
- flow sizes

For each parameter to be modeled, the distribution that fits the best our empirical traces is selected. Several distributions were considered, such as the Pareto, Lognormal, Poisson, Exponential, BiPareto, and Generalized Extreme value. Maximum likelihood was used for the parameter fitting while the evaluation of the goodness of the various distributions was performed using formal and visual statistical analysis methods and tools, such as the quantile plots with simulation envelopes. The following distributions model the user demand well:

- a **Time-varying Poisson** process models well the session arrivals at various APs in the infrastructure
- the **BiPareto** models well the flow size and number of flows within a session
- the **Lognormal** is a great candidate for the flow inter-arrivals within a session

The parameters of the distributions are based on empirical data that may correspond to different spatial and temporal scales. In increasing order of spatial scale, empirical data may be data collected from a specific AP, groups of APs located at the same building, groups of APs located at buildings of the same building-type, or all the APs in the infrastructure. The default time scale was the entire tracing period, but also finer time scales—such as daily and hourly—were also explored. The aggregation in the spatial and temporal domain trades the accuracy of the models for higher scalability and tractability. We first verified that the same distributions do persist across the aforementioned spatial and temporal scales. We then evaluated the tradeoffs between accuracy and scalability.

5.5.2 Models of user demand

This section illustrates our modeling approach considering the most aggregate spatio-temporal level, namely, the *system-wide* level. To fit the parameters of

the proposed models, it employs the entire trace collected from *all APs* of the infrastructure (i.e., system-wide approach).

System-wide approach

Although session arrivals vary widely, some expected patterns are apparent. Firstly, there is a clear diurnal periodicity, which is related to the substantial decrease of the network activity during the nights. Secondly, the activity of network clients decreases during the weekend. These temporal patterns appear to be common throughout the AP population, although some APs are more likely to be used at night than others.

The session arrival process is modeled as a time-varying Poisson process. We tested the validity of our modeling assumption with the statistical test described in Section 5.4.2. For the model to be valid, the variables R_{ijs} , which are defined in Eq. (5.2) as functions of the ordered session arrival times, must be exponentially distributed with a mean equal to unity and uncorrelated. The top part of Figure 5.14 shows an exponential quantile plot of the R_{ijs} during one randomly chosen hour.

We set the block length $L = 0.1$ hours in calculating the R_{ijs} . The quantile plot follows closely the diagonal line and remains well within the simulation envelope. This suggests that the exponential fit is clearly appropriate. The maximum likelihood estimate of the exponential parameter is 0.9372, which is very close to unity, and agrees with the claim that the R_{ijs} are standard exponential. The bottom plot of the figure plots the autocorrelations of the R_{ijs} up to 20 lags. The sample autocorrelations are always within the confidence intervals, so the R_{ijs} do not exhibit any significant correlations. Similar results were obtained when repeating the same analysis for other one-hour intervals of the 8-day dataset.

At the next modeling level, the arrival of a session triggers the arrival of a group of flows, initiated between the client and one or more Internet hosts. It is therefore natural to describe flow arrivals as a cluster process [278] rather than a point process in which flows arrivals are described in isolation. Since session arrival counts are (time-varying) Poisson distributed, flow arrivals form a cluster Poisson process. The flow-level traffic variables that need to be modeled with this approach are the number of flows associated to each session-cluster, and the inter-arrivals of flows within sessions. Our analysis showed that the BiPareto distribution yields the best fit for the number of flows per session. Figure 5.15 plots the complementary cumulative distribution function of the fitted distribution against the empirical data in a logarithmic scale. The circles are an equidistant set of samples from a BiPareto distribution with parameters $\alpha = 0.06$, $\beta = 1.72$, $c = 284.79$ and $k = 1$. The empirical distribution of the number of flows matches our model well for probabilities between 0 and 0.995. The fit is worse at the tail due to sampling artifacts. In any event, it is clear that the BiPareto model fits the empirical distribution very well. We also studied how the distribution of the in-session number of

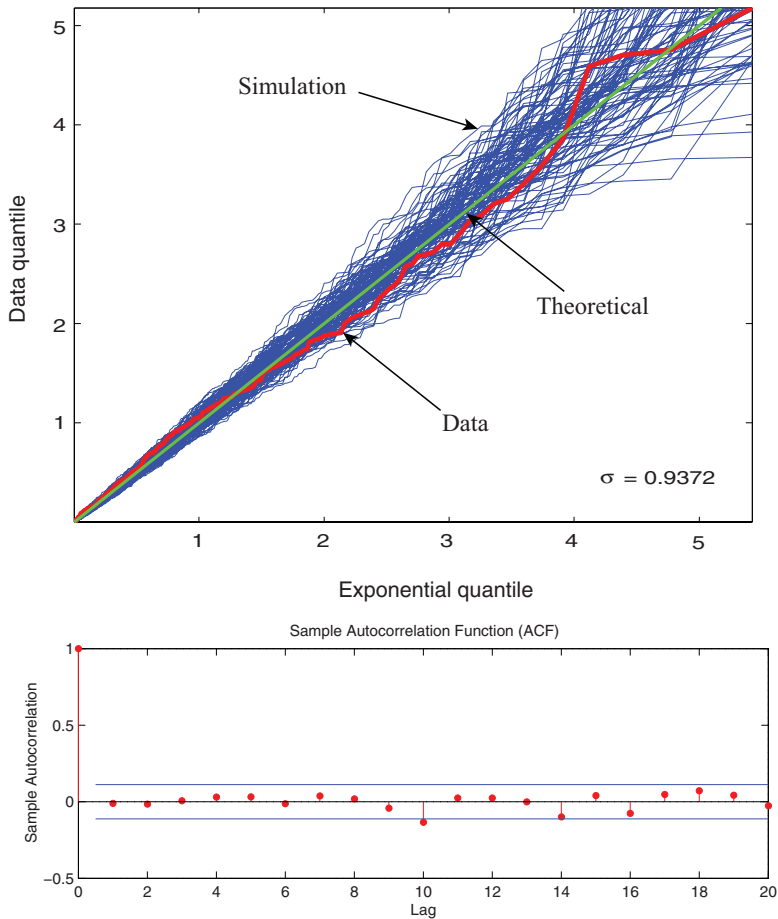


Fig. 5.14. The R_{ij} s are independent and exponentially distributed. Only one hourly block is shown here, but the results are consistent across the entire dataset.

flows varies per day. The distributions are very similar, with the vast majority of the sessions having between 1 and 1000 flows. The distributions for the weekends are slightly heavier. The number of flows per session goes as far as 10,000 for 0.1% of the sessions. This striking consistency of the curves strongly indicates that it is feasible to use parametric models for the traffic variables [217].

The second component of our cluster model is the distribution of the flow inter-arrivals within sessions. We showed that the Lognormal distribution provides the best fit, although the distribution is rather complex. The Lognormal

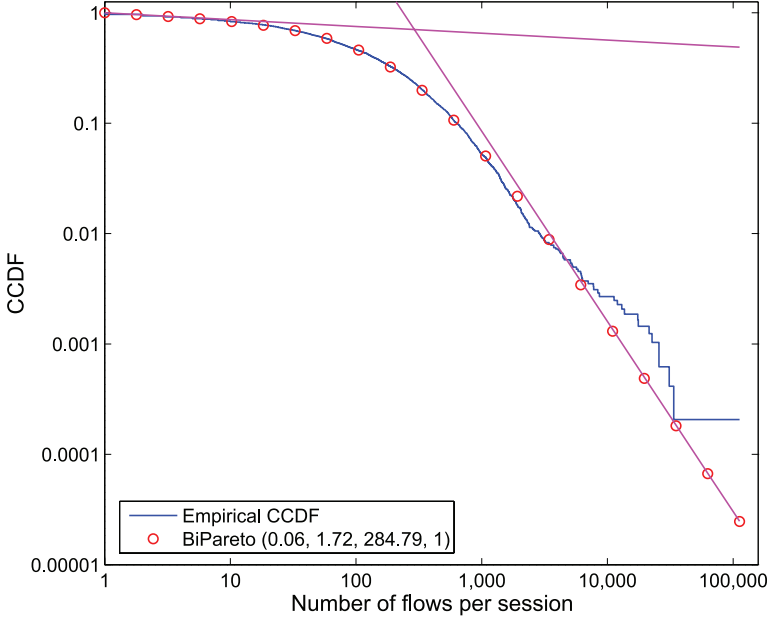


Fig. 5.15. CCDF for number of flows per session.

distribution appears to have similar shape to power law distributions. In a log-log plot of its CCDF, its behavior will appear to be nearly a straight line for a large part of the body of the distribution, especially when the variance of the corresponding normal distribution is large [266]. However, in contrast to power law distribution under natural parameters, a Lognormal distribution has finite mean and variance. The Lognormal quantile plot for the empirical data is shown in Figure 5.16; the parameters are estimated to be $\mu = -1.3674$ and $\sigma = 2.785$ using maximum likelihood. The quantile plot follows the diagonal line closely for all of the quantiles. The simulation envelope is very narrow in this case, and shows that some deviations from the Lognormal model in the upper part are significant. While more complex models, e.g., an ON/OFF model, may provide a better approximation, our Lognormal fit certainly provides a reasonable description of the data using only two parameters.

We have also studied the stationarity of the flow inter-arrivals within sessions and found that the flow inter-arrivals during each day are very consistent with each other [179]. To enable generation of traffic load in a manner suitable for experimentation, it is necessary to describe not only the flow arrival process but also the flow sizes in terms of number of bytes they transfer. Our statistical analysis reveals that flow sizes can be accurately described using

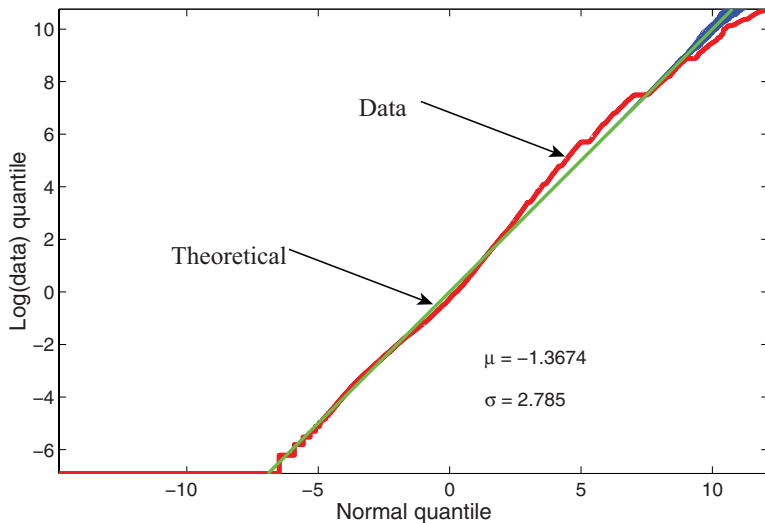


Fig. 5.16. Lognormal distribution for modeling flow interarrival.

a BiPareto distribution with parameters $\alpha = 0.00, \beta = 0.91, c = 5.20$ and $k = 179$. Figure 5.17 plots the BiPareto fit to the empirical data. The fit is excellent for most of the distribution with the BiPareto clearly capturing the transition in the slope between the body and the heavy tail of the empirical distribution. The approximation appears heavier than the empirical data at the end of the tail, which could motivate further refinements of the fit. We have also examined the stationarity of the flow size distributions over different days and found consistent tails considering all days in our tracing period, suggesting that weekly periodicities are not critical for modeling the flow sizes.

5.6 Syntrig: a synthetic traffic generator

Syntrig is a flexible synthetic traffic generator that obtains as input a set of distributions for the session arrival, number of in-session flows, flow interarrivals, and flow size. Based on these, it produces synthetic traces as shown in Figure 5.18. Specifically, it first produces a time series for the session arrival process, then samples the distribution of the number of flows to decide about the number of flows of the given session. Next, for these flows it selects the inter-arrivals (based on the corresponding distribution) to generate the within-session flow arrival time series. Finally, it assigns a size to each flow based on flow size distribution.

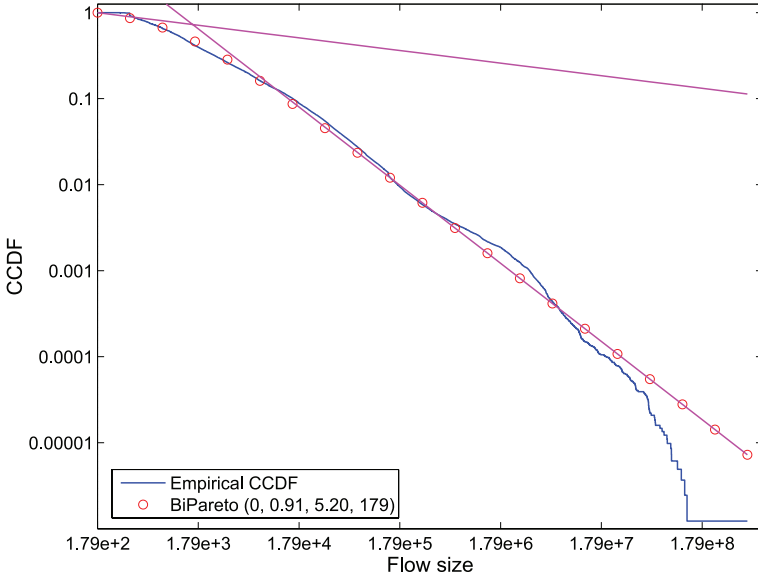


Fig. 5.17. Bipareto distribution formodeling flow size.

An emulation or simulation testbed (such as [369, 331]) can employ the generated synthetic traces as its wireless user workload.⁶ Syntrig’s input is a set of tunable parameters that are closely associated with the models of the session arrival, flow size, number of flows within a session, and flow interarrival times within a session. These parameters correspond to various conditions of the traffic load, application mix, and session profiles. By tuning Syntrig’s input, the produced synthetic traces “reflect” these conditions.

Each entry of the Syntrig output trace corresponds to a session and its associated flows. Specifically, it provides the following information:

- the session arrival timestamp
- the AP at which the corresponding session started
- the arrival of each in-session flow and its flow size

For fitting the parameters of the models, empirical traces—possibly at different spatio-temporal scales—are used. For example, for the generation of a synthetic trace at the AP-level, the empirical trace collected from that AP is used. Similarly, a synthetic trace at the “system-wide” (i.e., “network-

⁶ Packet-level details are left to the underlying protocols and are beyond the scope of this modeling effort (as explained earlier).

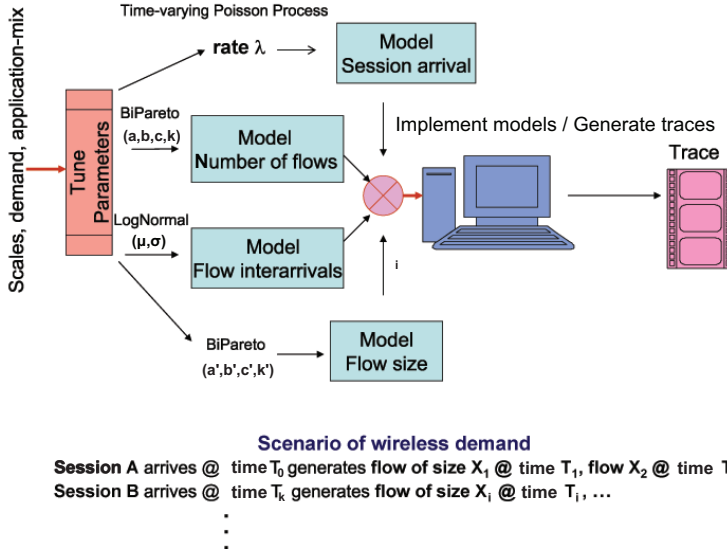


Fig. 5.18. Syntrig is a synthetic traffic generator that can produce synthetic traffic based on our models for various spatio-temporal scales, application-mixes, and workload.

wide”) level is based on the empirical traced collected from all APs in the infrastructure.

To produce the synthetic traces based on input models, the following steps were carried out:

1. For each hour of the corresponding empirical trace, the session arrival rate was estimated.
2. Synthetic session arrival times were produced at specific APs during that hour using the sessions arrival model.
3. For each session, its number of flows, flow inter-arrivals, and flow size values were generated based on the corresponding models.
4. Depending on the scale, the parameters of the input models were fitted using the corresponding empirical traces.

By tuning the spatio-temporal scales, application mixes, session profiles and rate, flow size, and flow interarrivals, Syntrig can produce various workload types. Such different workload types are useful in the context of capacity planning, admission control, and AP selection.

In general, Syntrig can be integrated with any type of model for the session arrivals, flow-size, flow interarrivals, and number of flows (e.g., traces for the

models described in Table 5.3 were produced using Syntrig). To produce synthetic traces based on the proposed models, the input included a time-varying Poisson session arrival model, a BiPareto distribution for the in-session number of flows and flowsizes, and a Lognormal for in-session interarrivals. Apart from the session arrival parameter that was *always* estimated based on the hourly building-specific empirical data, the parameters of all the other models were fitted using the empirical traces in the specified spatio-temporal scale.

Note that a simple transformation of the mean of the original Lognormal distribution for the flow interarrivals can produce a desirable new distribution. The increase of the flow size has an insignificant impact on the per-flow throughput. By multiplying the values of a BiPareto distribution by a factor, the resulting distribution is also BiPareto, with a scale parameter equal to the scale parameter of the previous one multiplied by a factor. Using this transformation, the Syntrig can tune the number of flows and flow sizes.

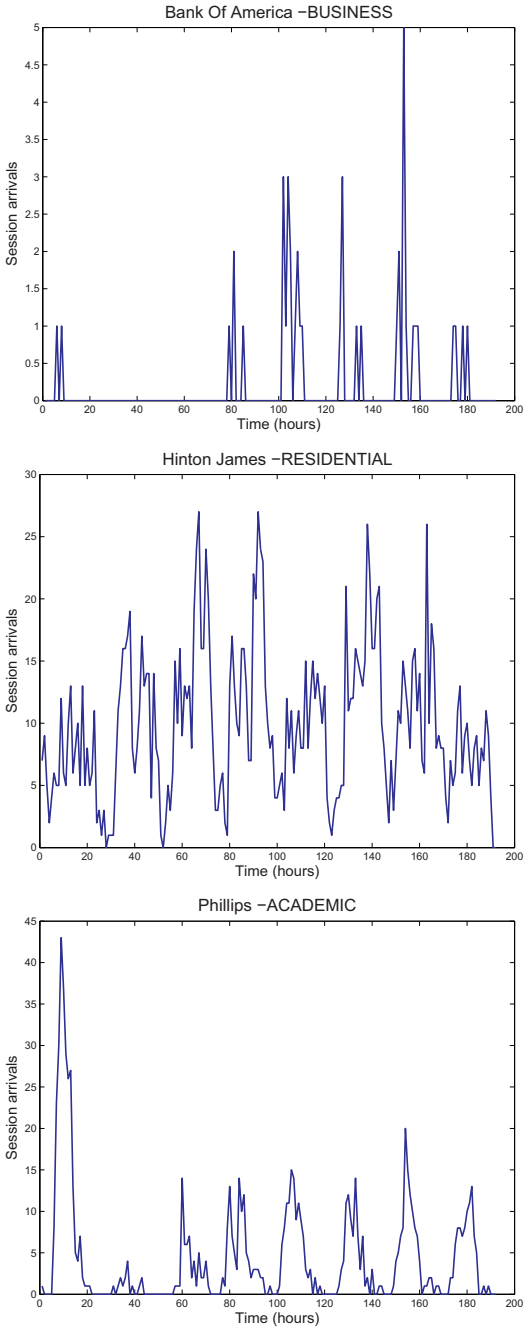
5.7 Scalability and reusability in user demand models

Scalability and reusability—properties particularly desirable in modeling—further complicate the modeling process. Previous modeling studies have either attempted to model traffic demand over hourly intervals at the level of individual APs [261] or studied the problem at the system-level, deriving models for the aggregate network-wide traffic demand, as in Section 5.5.2 [179]. Clearly, both approaches have their strong and weak points. The second approach results in datasets that are amenable to statistical analysis and provides a concise summary of the traffic demand at the system-level. However, it fails to capture the variation at a finer spatial detail that may be required for the evaluation of network functions with an emphasis on the AP-level (*e.g.*, load balancing). Despite these advantages when working at the AP-level, this approach fails in other respects. For example, it does not scale for large wireless infrastructures and the data does not always lend itself to statistical analysis. Moreover, the modeling results are highly sensitive to the specific AP layout of a particular network and the short-term variations of the radio propagation conditions.

The above challenges have motivated us to address the scalability and reusability tradeoffs. Our methodological choices attempt to strike a good trade-off between the two extreme approaches in traffic modeling that were outlined earlier:

- AP-level modeling
- infrastructure-wide modeling

To highlight the *spatial dimension of the variation*, we used *buildings* as basic entities for traffic demand modeling. Major features of user activity—such as traffic and roaming patterns—are studied at the *building level*, *i.e.*, group of APs located in the same building.



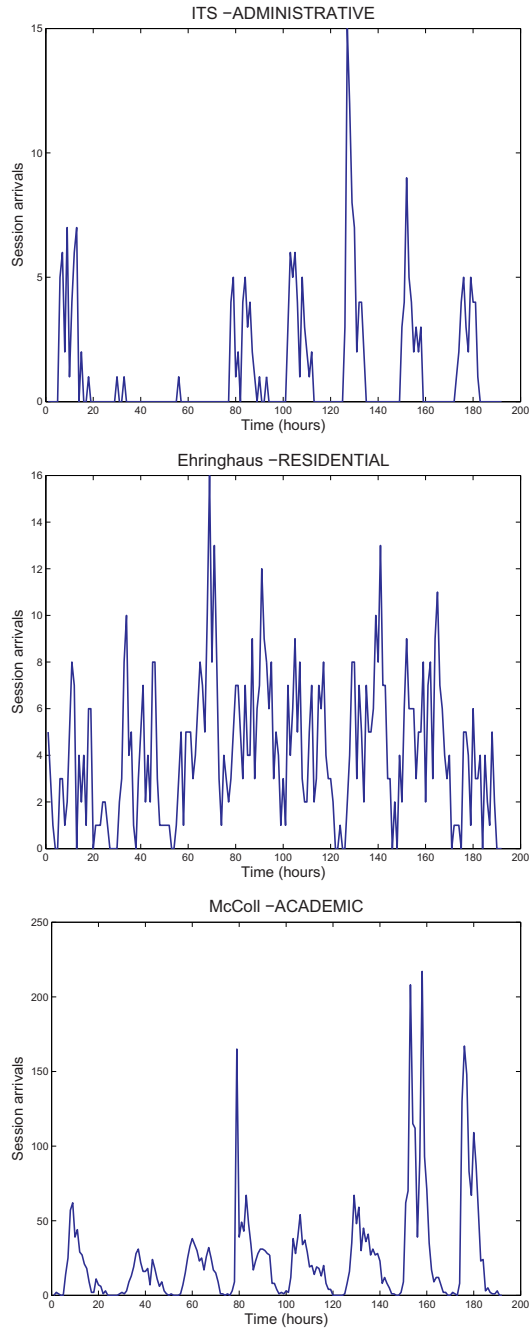


Fig. 5.19. Hourly session arrival rates for representative building types in UNC.

The spatial and temporal scales may increase from AP “AP” to building “BLDG” to building type “BLDGTYPE” to “NETWORK” or from day “DAY” to the entire tracing period “TRACE”, respectively. Let us indicate with the notation “A(B)” the scales of a modeling approach to be of spatial scale “A” and temporal scale “B”. The required number of sampling distributions for modeling each campus building under the “BLDG(TRACE)” approach would be $4N$, where N is the number of campus buildings. Repeating this procedure, for each single day of the trace (“BLDG(DAY)”) would increase this number by a factor of D , which denotes the number of days. When all buildings of the same type are modeled by a common set of distributions for flow-related variables, their number is reduced to $N + 3M$, where M is the number of building types. Smaller values of M can make the difference more dramatic and vice-versa. Thus, M acts as a tuning knob that can trade computing requirements with model accuracy and determines the complexity of the simulator. The type of building, the population of clients that access the network, the patterns of usage, and the environment are a non-exhaustive list of factors that contribute to the spatial and temporal variation of traffic demand. The following sections discuss how the modeled traffic variables vary across various time (hour, day, week) and spatial scales (building, building-type).

5.7.1 Variation of the session arrival rate within a day

Figure 5.19 plots the hourly session arrivals over the whole 2006 trace duration (192 hours) for some representative campus buildings. Although the absolute numbers of session arrivals and their exact variation are specific to each building, these profiles exhibit clear patterns that are, to a large extent, intuitive and closely related to the building type and usage. For example:

- Administrative and business buildings present clearly similar daily and weekly patterns in their profiles. The activity window is quite narrow during weekdays (6-8 hours long), in agreement with the working hours, whereas the activity during the weekend is almost zero.
- Residential buildings show distinctly different patterns. The number of session arrivals is more uniformly distributed across the week and hours within the day. The activity is also significant during the evening hours, often resulting in a daily or weekly peak.
- Academic buildings lie somewhere in between these two patterns. The daily window of activity is clearly broader than the administrative and business buildings, since they host WLAN clients for longer time intervals during the day. Weekends see fewer session arrivals and shorter windows of activity when compared with residential buildings, but traffic is non-negligible.

5.7.2 Variation of the session-level flow-related variables

The variation of traffic demand is also evident in the session-level variables. Their empirical distribution functions at the building-type level reflect this

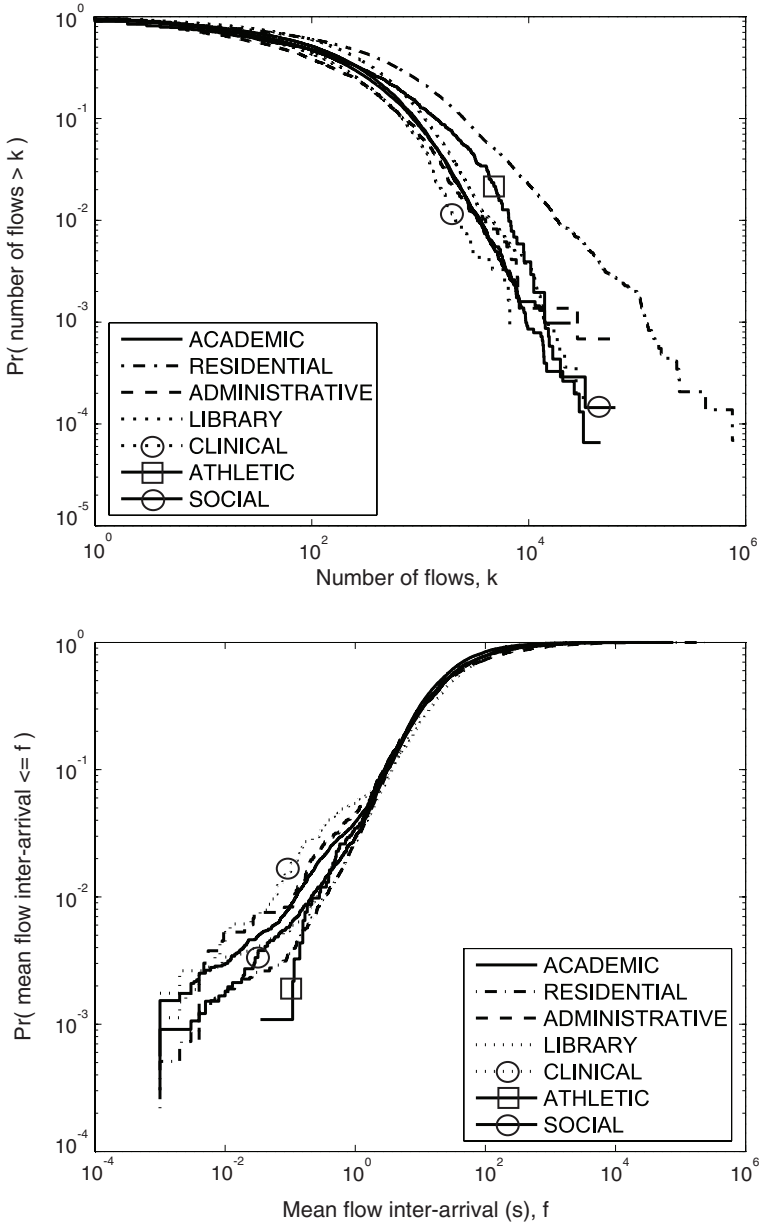


Fig. 5.20. Behavior of modeled session attributes across different types of campus buildings.

variation. Figure 5.20 (top) shows the broad variation of the per building-type distribution tails of the in-session number of flows. The number of flows related to the residential buildings sessions has a strikingly heavier tail, largely related to the more active web browsing behavior of residential users. The plots also suggest that the BiPareto distribution can be applied to model the per building-type in-session number of flows.

The behavior of flow inter-arrival times across different building types is presented in Figure 5.20 (bottom). Again, the plots of mean in-session flow inter-arrivals suggest that the variables could potentially be modeled by the same type of distribution for all building types, though with different parameters. The mean flow sizes across different building types are more similar [217].

The building type is an intuitive heuristic attribute for grouping buildings, providing a base for a unified treatment of the spatial dimension of the modeling task. The actual utility of this base is evaluated in the following section.

5.8 Evaluation of user demand models

This section evaluates our models in different spatio-temporal scales to highlight their accuracy and also addresses the accuracy and scalability tradeoffs using statistical-based and system-based metrics. These metrics were *not* explicitly addressed by our models. The time-varying Poisson session arrivals are *always* modeled using the hourly building-specific data. The main focus of this analysis is on the flow related parameters.

5.8.1 Statistical-based evaluation

The following statistics-based metrics are used:

- the **flow arrival** count process
- the **flow inter-arrival** time-series

The impact of various scales in the accuracy of our data is clearly illustrated in Figure 5.21. As either the spatial scale or temporal scale increases (from building “BLDG” to building type “BLDGTYPE” to “NETWORK” or from day “DAY” to the entire tracing period “TRACE”, respectively), the synthetic traces based on our models diverge from the empirical ones. A first view of the “noise” introduced by the aggregation is reflected in the deviation between the curves, as the spatial scale increases (e.g., “EMPIRICAL” compared to “BLDGTYPE(TRACE)”). The “EMPIRICAL” corresponds to the empirical trace collected from *all* APs in a *busy* building during the *entire tracing period*.

Staying at the building-type level does not result in a significant loss of accuracy compared to the building level. Despite its simplicity, the aggregation (“NETWORK(TRACE)”) does not result in substantially higher loss of information. Interestingly, the aggregation in the spatial scale may cancel out

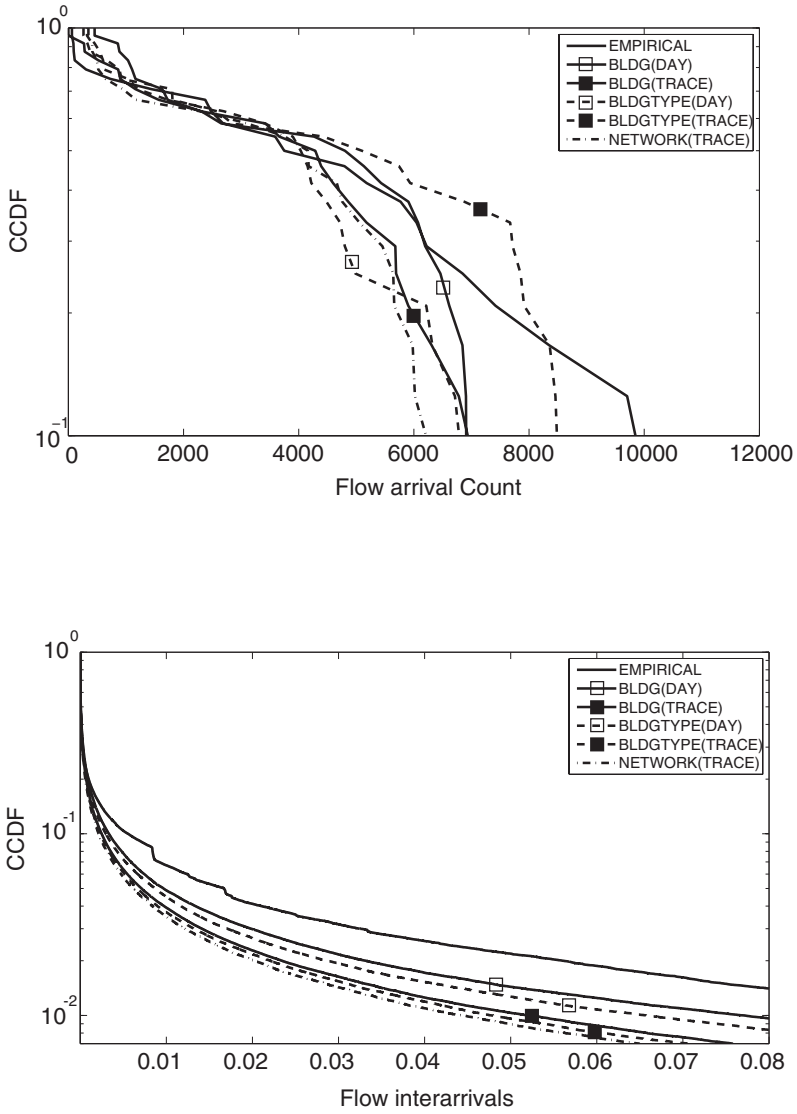


Fig. 5.21. Count of flow arrivals in an hour and flow interarrivals for different spatio-temporal scales.

the impact of the fine temporal scale (e.g., the performance of the “BLDG-TYPE(DAY)” compared to “NETWORK(TRACE)”).

Further improvement would be obtained by modeling the flow-related variables over shorter time periods than over the full monitoring period or over a day. In fact, the standard practice is to focus on modeling short-time windows where the building activity experiences its peak (busy hour).

5.8.2 Systems-based evaluation

When the statistical metrics show a deviation of the models from the empirical data, the systems-based metrics can be used to evaluate the impact of this difference on the performance of that system (or protocol). This section analyzes the performance of a hotspot AP under real-traffic conditions. As input for the user demand, it uses the empirical, real-life traces from UNC (EMPIRICAL) and synthetic traces based on our models. Furthermore, it performs a comparative analysis study of several models. The following metrics were employed to characterize the performance of the IEEE802.11 AP under real-life network conditions:

- *hourly aggregate* throughput
- *per-flow* delay, jitter, throughput, and goodput

Unlike throughput that takes into account *all* the data transferred in the transport layer, goodput *only* considers the amount of bytes delivered from the transport layer to the application layer. The delay per flow is the mean delay of a packet in the flow, which is the difference of the time required for the packet to be delivered at the receiver from the time it was enqueued at the sender. The jitter expresses the delay variability experienced by a receiver. The reported jitter value for a flow corresponds to the cumulative absolute difference between the delay of reception of consecutive packets.

It is expected that per-flow statistics will behave differently from the hourly aggregate statistics, given that most flows last less than one hour. Average per flow statistics are more sensitive to network dynamics than aggregate hourly flow statistics. The latter are less dependent on localized and transient phenomena, and can be useful to mechanisms (such as capacity planning, load-balancing, and admission control) that require knowledge of the user-demand in larger time scales.

The main objectives of this analysis are threefold:

- demonstrate the accuracy of our models using systems-based criteria
- highlight the impact of flow arrival and flow size on the throughput, goodput, jitter, and delay measured in a wireless LAN
- provide a comparative analysis study of various traffic models that simulate real-traffic demand conditions

Model	Size	Interarrival	Arrival
BIPARETO-LOGNORMAL	BiPareto	Lognormal	
BIPARETO-LOGNORMAL-AP	BiPareto	Lognormal	
PARETO-EMPIRICAL	Pareto	empirical	empirical
PARETO-UNIFORM	Pareto		uniform
FIXED-EMPIRICAL	fixed	empirical	empirical
EMPIRICAL-FIXED	empirical	fixed	
FIXED-UNIFORM	fixed		uniform
LOGNORMAL-WEIBULL	Lognormal	Weibull	
FIXED-FIXED	fixed	fixed	

Table 5.3. Generation of synthetic traces based on various models for the flow-based parameters. In some models (e.g., PARETO-UNIFORM), the flow arrival is modeled while in others (e.g., BIPARETO-LOGNORMAL), the flow inter-arrival. The fixed flow size is equal to the mean flow size in the empirical trace. The “empirical” in a parameter indicates an exact match of the values in the corresponding field of the synthetic and empirical traces.

Comparative analysis of various models

To illustrate the importance of accurate modeling of flow sizes and flow inter-arrivals in simulation studies, and also highlight the parameter with the greatest impact on the performance, we derived several additional models, summarized in Table 5.3. The following notation was used: “x-y” to indicate that the flow size follows the “x” distribution and the flow interarrival the “y”. Based on these models, synthetic traces were generated and replayed in the simulations. For fitting the parameters of these models, the empirical trace of a hotspot AP was used. Some of these models kept either the flow size or the flow inter-arrival identical with the corresponding data in the empirical trace (e.g., PARETO-EMPIRICAL and FIXED-EMPIRICAL). We experimented with flow arrivals that follow the uniform distribution and derived flow sizes from a Pareto distribution. Both distributions are popular choices for modeling the arrival process of flows, and the size of files downloaded via peer-to-peer applications, FTP, and HTTP [266]. The only difference between the PARETO-EMPIRICAL and FIXED-EMPIRICAL synthetic traces from the empirical trace is on the size of each flow. In particular, the PARETO-EMPIRICAL synthetic trace is based on flow size values derived from a Pareto distribution, while the FIXED-EMPIRICAL synthetic traces have flow size values that are fixed and equal to the mean flow size of the empirical trace. Notice that the total aggregate traffic of the FIXED-EMPIRICAL trace is the same as that of the empirical trace.

The flow arrival times of the PARETO-UNIFORM and FIXED-UNIFORM synthetic traces are derived from a uniform distribution on the interval $[0, T]$, where T is the duration of the empirical trace. The flow sizes in the PARETO-UNIFORM trace were derived using a Pareto distribution, while the FIXED-

UNIFORM trace includes flow sizes that are fixed, equal to the mean of the flow sizes in the empirical trace. The proposed models are BIPARETO-LOGNORMAL and BIPARETO-LOGNORMAL-AP. They are the only ones using the session abstraction, and the number of flows per session is modeled as a BiPareto distribution.

The LOGNORMAL-WEIBULL model (proposed by Meng *et al.* [261]) is composed of flow inter-arrival times that follow a Weibull distribution in an hourly basis and flow sizes that are based on a Lognormal distribution. The parameters of the Weibull distribution were determined using maximum likelihood estimation for *each hour-of-day* of the empirical trace. To fit the parameters of the Lognormal distribution for flow size, *all* flows of the empirical trace were used. In addition, synthetic traces based on naive models (e.g., FIXED-FIXED) were generated. In the FIXED-FIXED model, flow sizes are equal to the mean flow size of the empirical trace and flow interarrivals are equal to the mean duration of the mean in-session flow interarrival. Such models have been used extensively in performance analysis studies of wireless networking protocols.

All the synthetic traces were generated via Synt trig. To fit their parameters of all except the BIPARETO-LOGNORMAL model, the empirical trace corresponding to a *hotspot AP* was used. For the synthetic trace of BIPARETO-LOGNORMAL, the empirical trace of the *entire wireless infrastructure* was employed.

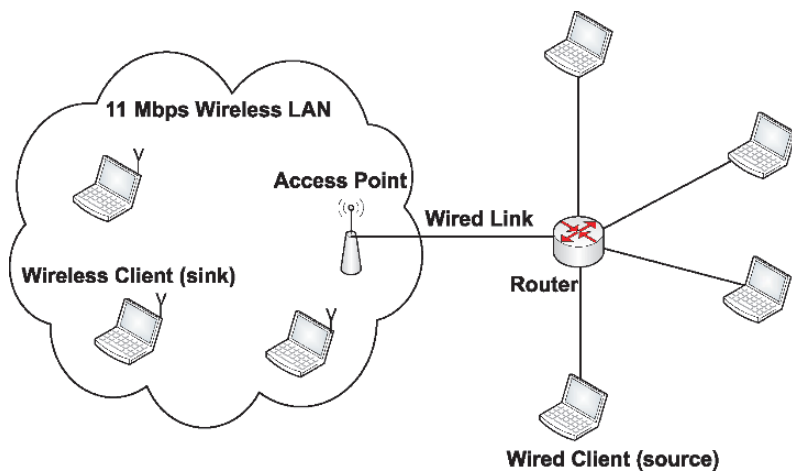


Fig. 5.22. The simulation/emulation testbed for analyzing the performance of a wireless LAN. The wired clients act as traffic “sources” (senders), while the wireless clients as “sinks” (receivers).

TCP-based experiments via simulations

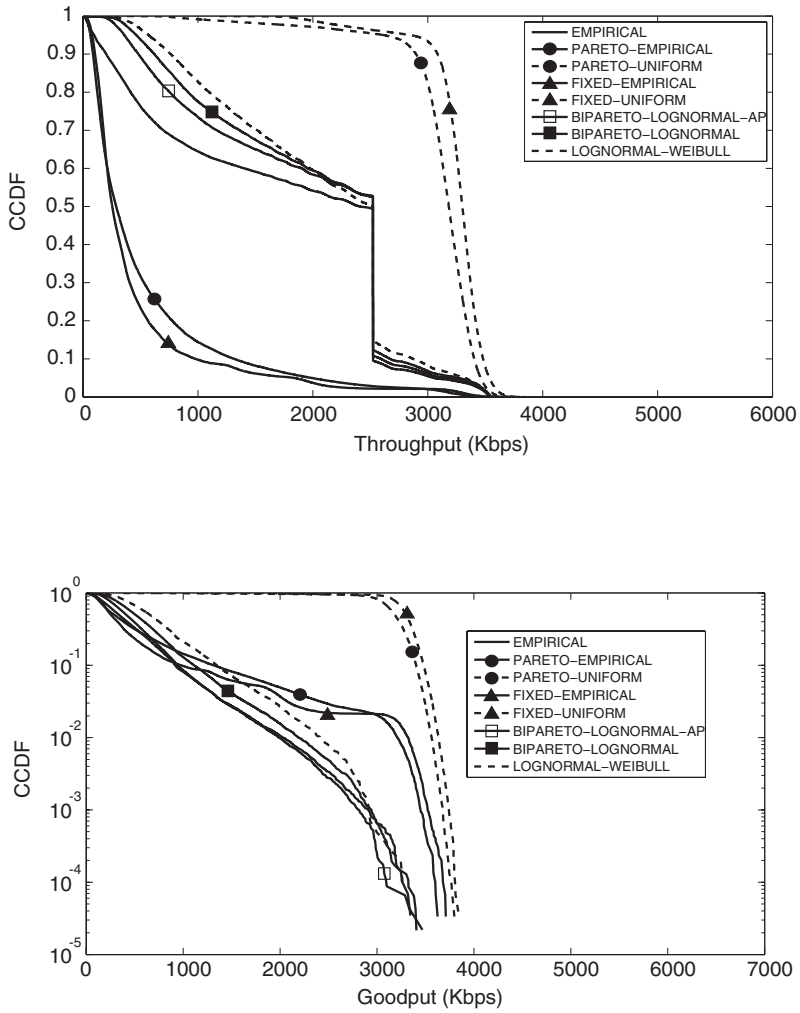


Fig. 5.23. Throughput and goodput per flow in a wireless hotspot AP simulated with real-traffic demand conditions.

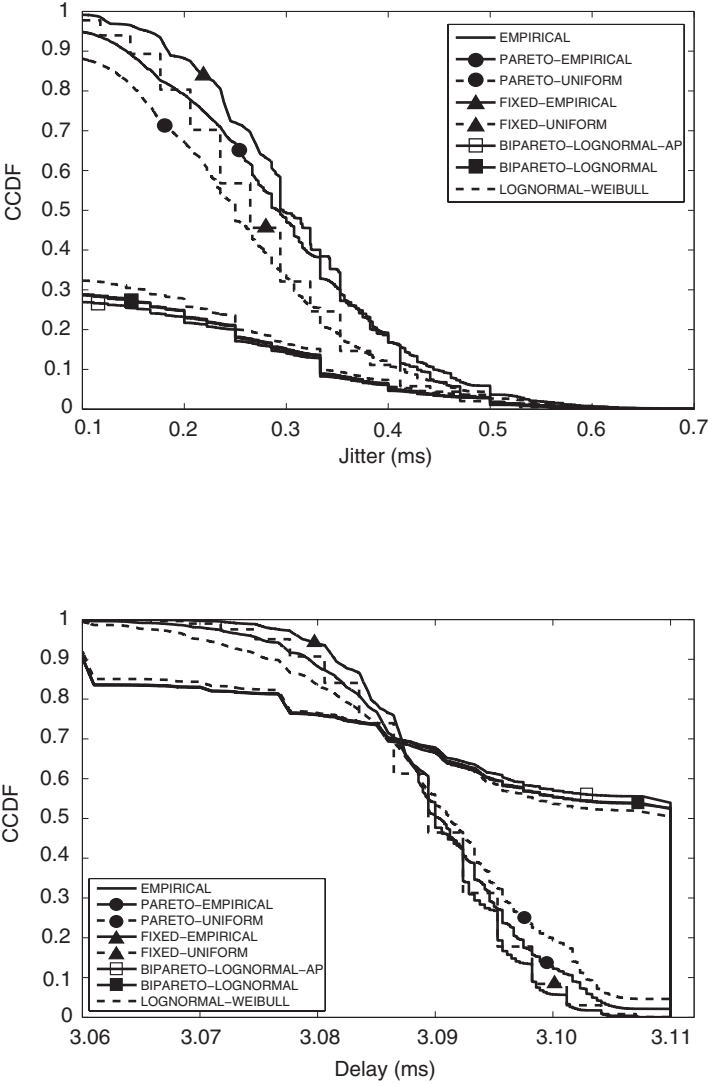


Fig. 5.24. Delay and jitter per flow in a wireless hotspot AP simulated with real-traffic demand conditions. The EMPIRICAL curve is very close to the BIPARETO-LOGNORMAL and BIPARETO-LOGNORMAL-AP.

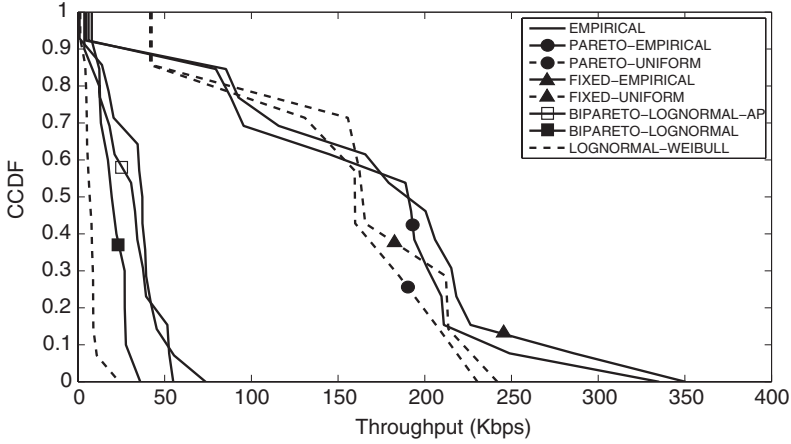


Fig. 5.25. Aggregate hourly throughput in a wireless hotspot AP.

The ns-2 testbed simulates a wireless LAN with three wireless clients associated with the same AP, and four wired clients connected via a router to the Internet (as shown in Figure 5.22). The link between the wired devices and the router has a speed of 100 Mbps. All wired links are duplex with a propagation delay of 2 ms, except for the one connecting the router to the AP which has a 1 ms delay, and a FIFO scheduling and drop-on-overflow buffer with a default size of 40 packets. The wired devices act as traffic sources, running an FTP application and sending data to the wireless clients (traffic sinks). The wireless clients use TCP Reno to download traffic from the Internet. Various synthetic and empirical traces were “replayed” in the simulation testbed. The session id “determines” the sink and each in-session flow is assigned to a source in a round-robin fashion.

A consistent trend for all benchmarks is that the BIPARETO-LOGNORMAL-AP model produces synthetic traces which when replayed in ns-2 result in a performance almost identical to the empirical ones (as shown in Figures 5.23, 5.24, and 5.25). The next best model, resulting in a performance close to the empirical traces, is the BIPARETO-LOGNORMAL. The LOGNORMAL-WEIBULL performs reasonably well. It should be reminded that the LOGNORMAL-WEIBULL trace was generated using empirical data collected for the *corresponding AP and specific hour of day*, and thus, is less scalable than the BIPARETO-LOGNORMAL one. Moreover, unlike the BIPARETO-LOGNORMAL and BIPARETO-LOGNORMAL-AP, it strongly underestimates the flow sizes.

The rapid drop in the throughput and delay per flow (in Figures 5.23 and 5.24) is due to the large percentage of flow sizes equal to the maximum segment size (MSS). These TCP flows correspond to transfers that carry less than

1 KB, and in ns-2, all the payload is packed in one MSS. In all the EMPIRICAL, BIPARETO-LOGNORMAL, BIPARETO-LOGNORMAL-AP, and LOGNORMAL-WEIBULL, a large percentage of flows with size of 1 KB or less was found.

The FIXED-FIXED model exhibits the worst performance among all these models.⁷ The departure of the PARETO-EMPIRICAL and FIXED-EMPIRICAL from the EMPIRICAL traces is prominent in both the hourly throughput and per-flow statistics and demonstrates the impact of the flow size. Note that although the FIXED-EMPIRICAL carries the same amount of total workload as the EMPIRICAL trace, its performance deviates substantially from the EMPIRICAL. Furthermore, the flow interarrival models have a prominent impact on the hourly throughput.

For the per-flow throughput and goodput, the flow inter-arrival exhibits a stronger impact than the flow size. For example, the FIXED-UNIFORM and the PARETO-UNIFORM models have similar performance. Likewise, the FIXED-EMPIRICAL and PARETO-EMPIRICAL models exhibit similar performance characteristics. When the distribution of the flow size remains the same while the flow interarrival distribution changes (e.g., PARETO-EMPIRICAL compared to PARETO-UNIFORM and FIXED-EMPIRICAL compared to FIXED-UNIFORM), their performance deviates prominently.

To demonstrate that the per-flow statistics can carry useful information for the performance of the network, we selected several hours from this hotspot AP with very close mean hourly throughput statistics and found that their per-flow throughput and delay statistics may differ substantially (Figure 5.26). The modeling study was repeated for hotspot APs with different application mixes, namely, an AP with 85% web traffic, a second one with 50% web and 40% peer-to-peer, and a third one with 80% peer-to-peer. The application-based classification was performed utilizing BLINC [300]. Using empirical traces from these APs, we fitted the parameters of our proposed models and produced the corresponding synthetic traces. Then, these traces were replayed in the simulation testbed. Figures 5.27 and 5.28 show clearly that for the first two APs with a large percentage of web traffic (50% or more), synthetic traces based on our models perform very similarly to the empirical traces. However, for the peer-to-peer traffic dominated AP, the performance of our models is less satisfying. Thus, the application mix can have a dominant impact on the accuracy of our models. Modeling the peer-to-peer traffic is not an easy task, especially due to the increased number, diversity, complexity and unpredictability in user interaction of peer-to-peer applications. Further analysis is required to investigate the robustness of our models under extreme network conditions (e.g., hours of highly congested networks, bad channel conditions, extensive mobility).

⁷ The performance of FIXED-FIXED deviates substantially from the others. The per-flow throughput and delay is constant and approximately 3.2 Mbps and 3.0857 ms, respectively.

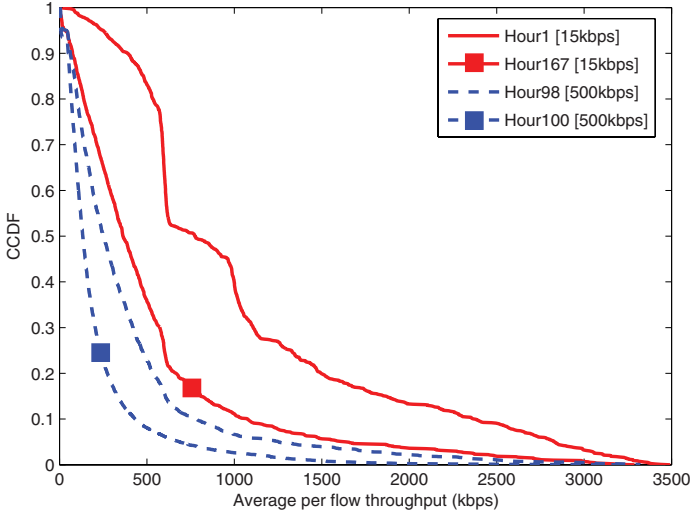
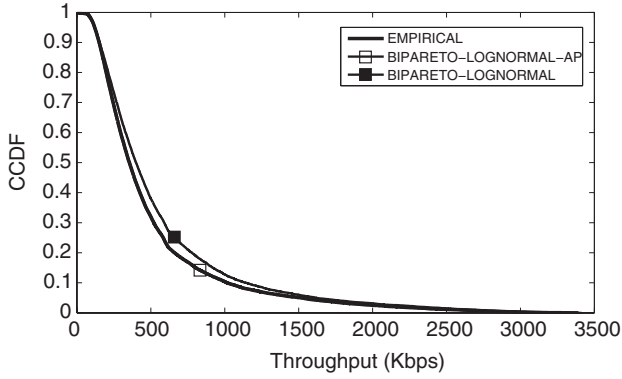


Fig. 5.26. Comparing per-flow statistics for hours that have produced the same aggregate download traffic.

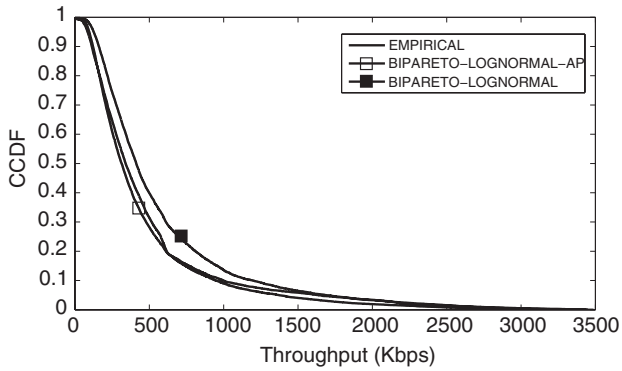
Emulations on TCP

Often simulations fail to capture all the interactions and dependencies across the different layers. Emulations can be used to provide a more thorough look over the controlled experiments and further validate the performance results. For this purpose, we repeated the study on a small testbed using Harpoon [331].⁸ The emulation testbed consists of three stationary wireless devices—operating as network sinks—a desktop PC—running all three corresponding Harpoon servers—and a Cisco Aironet 1200 AP, operating in IEEE802.11b. The servers replayed our empirical and synthetic traces. The wireless clients and the server were synchronized using NTP. Packet transfers on transport layer were monitored by tcpdump, running on all four computers. Using these packet header traces, we measured the performance of the AP based on the same benchmarks as in simulations. As in simulations, the synthetic traces based on our models resulted in a performance very close to that when empirical traces were used.

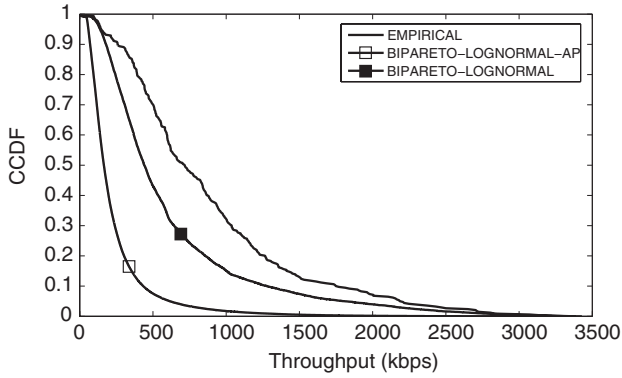
⁸ Harpoon can be used in an emulation testbed to generate flows with certain flow arrival and flow size values provided as input distributions.



(a) hotspot with 85% web traffic

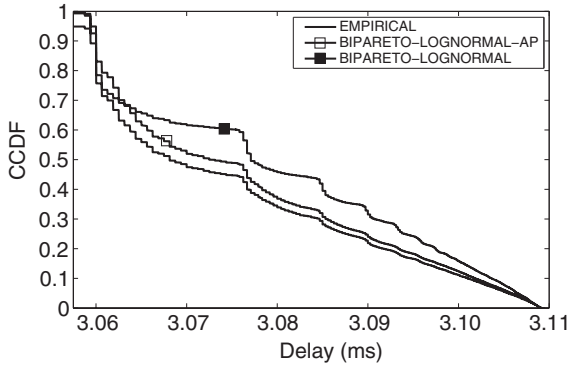


(b) hotspot with 50% web and 40% peer-to-peer traffic

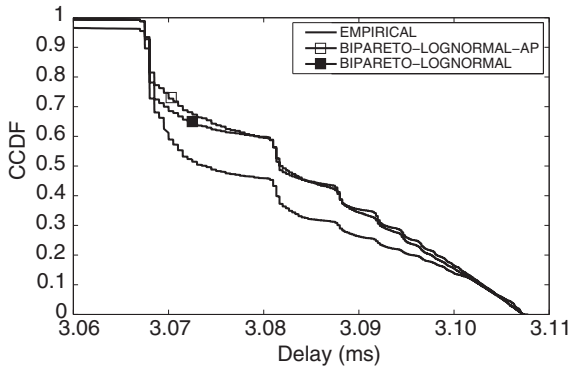


(c) hotspot with 80% peer-to-peer traffic

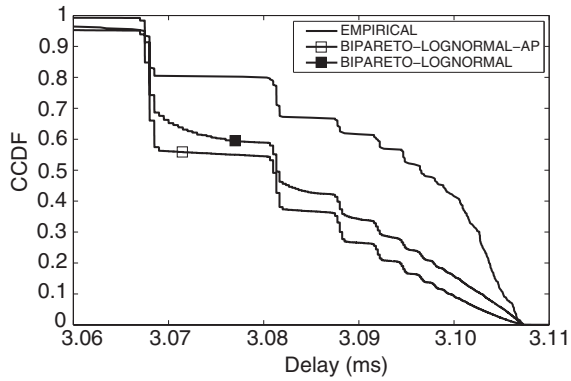
Fig. 5.27. Impact of the application mix on per-flow throughput (selected hotspots with different application mixes).



(a) hotspot with 85% web traffic



(b) hotspot with 80% web and 40% peer-to-peer traffic



(c) hotspot with 80% peer-to-peer traffic

Fig. 5.28. Impact of the application mix on per-flow delay (selected hotspots with different application mixes).

UDP-based experiments via simulations

To further evaluate the models, UDP-based scenarios were performed. In order to do this, the BIPARETO-LOGNORMAL-AP, BIPARETO-LOGNORMAL-AP, EMPIRICAL, and CBR-based traffic (popular in simulation studies) were compared with respect to hourly throughput. In the simulation experiments, the number of communicating pairs (wired senders and wireless receivers) were drawn from a uniform distribution in the range of [2,10]. The amount of data replayed in each set of experiments was equal to the aggregate traffic in the original trace. A small random delay was introduced in the arrival time of each flow to avoid the concurrent start of all flows. Each session of the BIPARETO-LOGNORMAL-AP, BIPARETO-LOGNORMAL-AP, and EMPIRICAL traces, is “assigned” to a pair (wired sender and wireless receiver). A UDP transmission is then initiated of size equal to the defined flow size at the specified flow arrival time, for each flow. In CBR scenarios, each source transmits a persistent UDP flow with size equal to that of the original trace divided by the total number of pairs. For the CBR simulations, the mean computed from a total of ten runs was reported (Table 5.4). The BIPARETO-LOGNORMAL-AP-, BIPARETO-LOGNORMAL-, and EMPIRICAL-based scenarios perform similarly, deviating from the CBR-based ones.

CBR rate	Median throughput
10 Kbps	9.7 Kbps
25 Kbps	15.2 Kbps
50 Kbps	30.5 Kbps
100 Kbps	57.7 Kbps

Table 5.4. Median hourly throughput in scenarios using CBR sources.

Table 5.4 shows statistics on the hourly throughput for various CBR transmission rates. When wired clients transmit at 25 Kbps, the mean hourly throughput is 13.2 Kbps. The mean hourly throughput in the empirical trace is 1.7 Kbps, while in the BIPARETO-LOGNORMAL-AP and BIPARETO-LOGNORMAL, it is 10.3 Kbps and 9.7 Kbps, respectively. The median hourly throughput in the empirical trace is 1.4 Kbps, while in the BIPARETO-LOGNORMAL-AP and BIPARETO-LOGNORMAL, it is 2.4 Kbps and 2.6 Kbps, respectively. On the other hand, the median hourly throughput is 15.2 Kbps. Thus, the traffic based on the CBR models results in an hourly throughput distribution that differs *substantially* from the one produced using the empirical traces.

5.9 Singular spectrum analysis of traffic at APs

For quality of service provision, capacity planning, load balancing, and network monitoring, it is critical to understand the traffic characteristics. For this

purpose, the analysis of the traffic load time-series at APs can be important. In some earlier studies, we modeled the traffic load at APs using variants of the Moving Average and Autoregressive Moving Average models [287, 282].

Due to the complicated structures of these traffic load series, traditional algorithms of non-linear analysis may not result in reliable estimates. However, after filtering out a high-frequency component—which can be considered as a noisy part—we could expect to obtain a more accurate estimation of the embedded dimension of the underlying process. Motivated by this observation we analyzed traffic series by decomposing them in two components, namely, a low-frequency and a high-frequency one, using Singular Spectrum Analysis (SSA).

SSA [162] belongs to the general category of PCA methods [208]. The SSA method is very effective in the analysis of time-series corresponding to an arbitrary process. In a recent work [63], SSA was used to analyze the dynamics of traffic obtained in an intermediate-scale wired LAN. To the best of our knowledge, our study in [343] is the first one that applies SSA to the analysis of traffic from a WLAN.

SSA allows us to explore the intrinsic dimensionality and structure of the time-series corresponding to the traffic load at a given AP, using data collected from a campus-wide WLAN infrastructure. To investigate the nature of this dimensionality, we introduce the notion of *eigenloads*. Derived from the implementation of SSA on a given traffic load series, an eigenload is a time-series that captures a particular source of temporal variability. Each traffic load series can be expressed as a weighted sum of eigenloads, where the weights are proportional to the extent to which each eigenload is present in the given traffic load series.

We show that traffic eigenloads in a WLAN fall into two natural classes:

- *deterministic eigenloads*, which capture the slow-varying trends in the traffic load series
- *noise eigenloads*, which account for traffic fluctuations appearing to have relatively time-invariant properties

By categorizing eigenloads in this manner, we can obtain a significant insight into the intrinsic properties of the traffic load series. Our main findings can be summarized as follows:

- Each time-series can be well approximated by only a small number of eigenloads, which constitute its “feature set”.
- These features vary in a predictable way as a function of the amount of traffic carried in the time-series.
- The largest traffic load series, i.e., the series with the highest mean traffic load, are primarily deterministic. On the other hand, traffic load series of moderate size are generally comprised of noisy features.

Motivated by the observation that the deterministic part of a traffic load series presents a slow variation in time and carries the main part of the information

content, we designed a predictor that performed trend forecasting at a larger than an hourly time-scale. This forecasting algorithm is based on the modeling of the traffic-series using a linear model of order p , whose coefficients (weights) were estimated using the Normalized Least Mean Squares approach.

In future work, we will complete the design of the proposed predictor by taking into account not only the deterministic, but also the noisy component of a given traffic series. For this purpose, an optimal radial basis function will be trained for the prediction of the noisy part [244].

Another interesting problem is the detection of dynamic changes of the future traffic load values. In particular, the accurate detection of transitions from a normal to an abnormal state, either due to hardware or software failure, or due to an attack, may improve diagnosis and treatment. The multi-scale decomposition given by the SSA approach, could be combined with the conceptually simple and computationally very fast concept of permutation entropy [103] to detect dynamical changes in the subset of noisy eigenloads, which are responsible for the transient behavior of the traffic load series.

5.10 Related work

A large body of literature has developed concepts and techniques for modeling Internet traffic, especially in terms of statistical properties (e.g., heavy-tail, self-similarity). For example, heavy-tailed distributions appear in the sizes of files stored on web servers [124], data files transferred through the Internet [294], and files stored in general-purpose Unix filesystems, suggesting the prevalence and importance of these distributions. Self-similarity characteristics exist in Internet traffic. In a pioneering work, Leland *et al.* showed that LAN traffic exhibits a self-similar nature [243]. Evidence of self-similarity was also found in WAN traffic [296]. In that work, Paxson and Floyd demonstrated that self-similar processes capture the statistical characteristics of the WAN packet arrival more accurately than Poisson arrival processes, which are quite limited in their burstiness, especially when multiplexed to a high degree. Self-similar traffic does not exhibit a natural length for its “bursts”. Its traffic bursts appear in various time scales [243]. The relation of the self-similarity and heavy-tailed behavior in wired LAN and WAN traffic was analyzed by Willinger *et al.* [356]. On the other hand, Poisson processes can be used to model the arrival of user sessions (e.g., TELNET connections and FTP control connections). However, modeling packet arrivals in TELNET connections by a Poisson process may result in inaccurate delay characteristics, since packet arrivals are strongly affected by network dynamics and protocol characteristics.

Web traffic exhibits also self-similarity characteristics. Crovella and Bestavros showed evidence of this and attempted to explain them in terms of file system characteristics (e.g., distribution of web file size, user preference in file transfer, effects of caching), user behavior (e.g., “think time” accessing a web

page), and the aggregation of many such flows in a LAN [123]. The majority of web traffic in wired networks is below 10 KB while a small percentage of very large flow account for 90% of the total traffic. They employed power-laws to describe web flow sizes. We also observed similar phenomena in the campus-wide wireless traffic. A nice discussion of the use of power law and lognormal distributions in other fields can be found in [266].

Peer-to-peer applications evolve rapidly, dominating the traffic mix in several cases. As recent studies have indicated, peer-to-peer and web traffic differ significantly (e.g., unlike in web, where web clients may download a popular web page, multiple times, the immutability of Kazaa's multimedia objects leads clients to fetch objects at most once) [168]. However due to their increasing number, the differences in their communication pattern, and the difficulty to classify them accurately, modeling of peer-to-peer traffic is challenging.

Two general approaches for traffic generation are the *packet-level replay* and *source-level* generation. The packet-level replay is an exact reproduction of a collected trace both in terms of *packet arrival times*, size, source and destination, and content type. To analyze a system under various traffic conditions, researchers need to employ the appropriate packet-level trace that exhibits the required traffic conditions. However, a packet-level trace is directly affected by the network conditions under which it was generated. Collecting the appropriate empirical data is a non-trivial task. Specifically, reproducing the intended packet arrival process can be complex due to the arbitrary delays introduced at the various network components by various interrupts, service mechanisms, and scheduling processes. On the other hand, closed-loop or feedback-loop characteristics manifest the reactions of the source and destination of a flow to network conditions, triggering further changes (e.g., TCP's congestion avoidance mechanism). However, packet-level replays cannot reflect such feedback-loop characteristics.

Adopting a different approach, the *source-level* models the *sources* of traffic (e.g., the applications running on the source and destination). These sources are used as building blocks, along with the various network components that can be modeled or simulated, allowing the analysis of a system under various conditions. The generation of packet-level data can be based on some statistical properties that characterize the empirical data, and thus, ensure that the synthetic data are "realistic enough". However, it is important to note that the realism of a trace depends tightly on the system to be studied. The selection of these statistical properties that are general enough but also tunable to express different traffic conditions/profiles is a non-trivial task and depends on the characteristics of the system to be studied. The source-level approach, advocated by Paxson and Floyd [149], allows the underlying network, protocol, and application layer to specify and control the packet arrival process. The infinite source model is one of the simplest and popular source-level models. It has no parameters and is used to model very large network flows. However the infinite source model models the traffic poorly, since the majority of Internet traffic is relatively light, with bidirectional flows, and of small packet

size [107, 211, 150]. An enlightening discussion of these approaches is included in [178].

Our approach is inspired by the source-level (or network independent) modeling. The main assumption is that session arrivals—initiated by humans—at a large extent are not affected by the underlying network technology. Furthermore, given the relatively low percentage of packet loss at the network layer, we assumed that the in-session flow size and flow arrivals can approximate the *intended* user traffic demand. The proposed user workload traces can then be integrated with a channel, packet generation, and network topology model to simulate/emulate certain conditions in the context of a performance analysis study.

Traffic generation is an important aspect of the network modeling and simulations. Several studies have addressed the challenges and provided guidelines on generating realistic synthetic traffic in wired networks [296, 149, 178]. In general, traffic generators may either use mathematical models (e.g., a Poisson process) or empirical data (e.g., Swing [350]). Swing focuses on characterizing and mimicking packet inter-arrival rate, packet size distribution, destination IP address, and port distribution in a wired network. Unlike Swing that aims to produce synthetic traces that capture the network conditions, our objective is the generation of user demand traces based on accurate models of intended traffic demand, independent from specific network characteristics. Using our proposed models of the intended traffic workload, synthetic packet-level traces can be generated based on various application-mix, channel, link, and mobility patterns. Depending on the protocol of interest, it is important to select the appropriate metrics to validate that the generated packet-level trace reflects the desirable network conditions. Examples of such metrics are the burstiness of the packet arrival process at various time-scales, packet size distribution, and distribution of send and receive traffic. However, the generation of synthetic packet-level traces to reflect certain network conditions by tuning the appropriate parameters in a simulation or emulation testbed (e.g., buffer size, interference, traffic load, link and channel characteristics) is a hard problem.

While there is rich literature on traffic characterization in wired networks (e.g., [355, 80, 120, 102, 278]), there is significantly less work of the same depth for WLANs. Hierarchical approaches to modeling the wireless demand and its spatial and temporal phenomena have received little attention from our community. In fact, the only relevant study we are aware of is the flow-level modeling study by Meng *et al.* [261]. The authors used the available Dartmouth traces, that include SYSLOG messages and TCPDUMP data from 31 APs in five buildings. They proposed a two-tier (Weibull regression) model for the arrival of flows at APs and a Weibull model for flow residing times, and they also observed high spatial similarity within the same building. The authors also studied the modeling of flow size, and suggest that a Lognormal model provides the best approximation.

Minkyong *et al.* [223] clustered APs based on their peak hours and analyzed the distribution of arrivals for each cluster, using the aggregate client

arrivals and departures at APs. Similar clusters based on registration patterns were also reported by Ravi Jain *et al.* in their modeling study of user registration at APs [201].

5.11 Conclusions

We introduced a novel methodology for modeling the wireless access and traffic demand by providing a multilevel perspective. In particular, we modeled the arrival and size of sessions and flows considering various spatio-temporal scales and explored their statistical properties, dependencies and inter-relations. Time-varying Poisson processes provide a suitable tool for modeling the arrival processes of clients at APs. We validated these results by modeling the visit arrival rates at different time intervals and APs. In addition, we proposed a clustering of the APs based on their visit arrival and the functionality of the area in which they are located. The models have been validated using empirical data from different *time periods* (an entire week in April 2005 and another one in April 2006), different *time scales* (week, day, hour), different *spatial scales* (AP, group of APs located within the same building, set of APs located within buildings of the same functionality, and entire wireless infrastructure), and various application-mixes. The BiPareto distribution models well the flow sizes of the Dartmouth trace, collected from its wireless campus-wide infrastructure.⁹

Although the absolute numbers of session arrivals and their exact variation are specific to each building, these profiles exhibit clear patterns that are, to a large extent, intuitive and closely related to the building type and usage [217]. Also, the mean in-session flow inter-arrivals across different buildings and building types suggest that the variables could potentially be modeled by the same type of distribution for all building types, though with different parameters. The mean flow sizes across different building types are very similar [217]. Furthermore, the empirical traces collected from APs with 50% or more of web traffic can be fitted nicely by the proposed models. However, for workloads dominated by peer-to-peer traffic, the fit deteriorates significantly.

Syntrig generates synthetic traces based on a set of tunable parameters (i.e., its input). These parameters are tightly associated with the proposed models and can reflect various conditions, such as flow sizes, flow interarrival times, session arrivals, application mixes, and session profiles. The obtained synthetic traces can then be “replayed” in emulation or simulation testbeds in the context of a performance analysis study. Synthetic traces based on our models result in a performance very close to the one when empirical traces are used as input. Furthermore, synthetic traces based on popular models—employed frequently in simulations—exhibit large deviations from the empiri-

⁹ Given that session-related information could not be generated using the available Dartmouth traces, only the flow size models were validated.

cal traces. The trade-offs between accuracy and scalability of our models were evaluated using statistics-based and systems-based benchmarks.

Different synthetic traces can be generated for various application mixes, traffic loads, and user profiles. Such traces can be used in the performance analysis of algorithms for capacity planning, dimensioning, and admission control under different traffic load, user profiles, and application mix conditions. The flexibility in defining different profiles is desirable, especially given the fact that the user traffic demand cannot be easily determined: new applications and services gain popularity and new user behavior, type of devices and access patterns emerge. Thus, a natural next objective is to derive client profiles, a more intuitive abstraction than session profiles. Understanding this part of the workload will make simulations more intuitive, in the sense that the input could be the number of clients and perhaps some parametric description of their long-term access patterns. Ideally, these client profiles would be based on the proposed session and flow models. Two other important objectives are to evaluate the use of the proposed models to *predict* the traffic demand, and to obtain a better understanding of the impact of the “heavy” tails of the in-session flow size, number of flows, and flow interarrival distributions on the design of various wireless networking mechanisms.

It would be interesting to explore different user workload profiles, utilizing traces from emerging wireless environments. As more traces from various wireless network environments become available, it is critical to develop methodologies and tools for searching for “law-like” relationships across these different traces that can be generalized to a wide range of different conditions.

Conclusions and future work

6.1 Conclusions

The advances in wireless communications and the adoption of mobile computing devices have further impelled the evolution of pervasive computing space. We envision users with wirelessly-enabled devices, interacting with such pervasive computing spaces to access, generate and share information, forming new social networks and networking paradigms. In such networking environments, self-organizing, autonomous devices interact with each other to enhance information access. Their autonomy, self-organization, and cooperation led us to explore the peer-to-peer paradigm.

Our research was driven by several questions: Given their frequent disconnections, how can wireless devices exploit their increasing storage and processing power to enhance the information access? How fast does information diffuse in such mobile networks? Does wireless access exhibit high spatial locality of information? What is the interplay between device cooperation and data availability? What are the gains if devices act as miniature mobile caches? How do clients access wireless networks and what is their traffic demand?

6.1.1 Mobile peer-to-peer computing

We proposed 7DS, a novel mechanism that enables wireless devices to share resources in a self-organizing manner, without the need of an infrastructure. In information sharing, peers query, discover, and disseminate information, while for message relaying, hosts forward messages to the Internet on behalf of other hosts when they gain Internet access. The percentage of hosts that acquire the data object as a function of time and their average delay were measured. We found that the density of the cooperative hosts, their mobility, and the transmission power have the most pronounced impact on data dissemination. The synchronization of the periods that the network interface of peers is powered and the reduction in the frequency of querying can save

energy. In the case of FIS with a low density of hosts, the query frequency can be set as large as three minutes without impacting on the speed of data dissemination. Similar results hold in the case of P-P.

The performance of data dissemination remains the same when the area is expanded but the density of the cooperative hosts and the transmission power are kept fixed. Also, for a fixed wireless coverage density, the larger the density of cooperative hosts, the better the performance. In S-C, this implies that for the same wireless coverage density, it is more efficient to have a larger number of cooperative hosts with lower transmission power than fewer with a higher transmission power. We also presented an analytical model for FIS using theory from random walks and environments and the kinetics of diffusion-controlled processes.

The spatial locality of information was the driving force behind 7DS. To evaluate the degree of spatial locality in a real environment, we analyzed web requests collected from a large-scale wireless network. Although the web is not primarily a location-dependent or collaborative application, its prevalence motivated this analysis. The spatial locality can be computed for various spatial granularities. We mostly concentrated on AP-, building-, and infrastructure-wide levels. Specifically, we measured how likely it is for two peers co-resident within an AP to be interested in the same data, and how likely it is for a client to request a data item that is already stored in the AP-, building-, or infrastructure-wide level cache. The building-level cache is an aggregation of all the caches of APs located in that building, while the infrastructure-wide cache is an aggregation of all the caches of all the APs in the infrastructure. The following caching paradigms were analyzed:

- user cache
- cache attached to an AP
- peer-to-peer cache, in which peers are devices associated with the same AP
- campus-wide cache

The overall ideal hit ratios of the user cache, cache attached to an AP, and peer-to-peer caching are 51%, 55%, and 23%, respectively. The ideal hit ratio across APs varies and was found to be as high as 73%. For such APs, a local AP cache can be beneficial. In general, the spatial locality of the wireless web access varies across APs. Wireless web access also exhibits high temporal locality. Each client frequently requests objects that it has requested within the past hour, and occasionally, requests objects that have been requested by other nearby users within the past hour.

We also applied the peer-to-peer paradigm to positioning for mobile computing devices. Our proposed system, CLS, positions wirelessly-enabled devices using the existing wireless communication infrastructure adaptively without the need of specialized hardware or training. To improve its accuracy, CLS enables hosts to cooperate and share positioning information and

also allows the integration of external information, such as maps, popular routes, and user mobility patterns.

6.1.2 Wireless measurements and modeling

In general, networks are extremely complex and the interaction of different layers and technologies creates many situations that cannot be foreseen during the design and testing stages of technology development. This is especially true for wireless networks, which are used for many different purposes, and which are based on a shared medium that is inherently more vulnerable than its wired counterpart. One of the lessons learned during this research was that it is critical to perform measurement-based studies, in order to uncover deficiencies and identify possible optimizations for better utilizing the scarce resources in wireless systems. As mentioned earlier, a typical evolution of a technology consists of the following steps:

1. simple simulations
2. advanced and more realistic simulations
3. emulations and tests in small-scale testbeds
4. tests in large-scale testbeds
5. adoption and use in real-life environments

The existence of testbeds, tools, benchmarks, and models is of tremendous importance and can be a catalyst for further performance analysis and simulations.

Wireless networks have their own distinct characteristics and challenges due to the radio propagation characteristics and mobility. Some typical assumptions in performance analysis studies on wireless networks are the following [100, 364]:

- models and analysis of wired networks are valid for wireless networks
- wireless links are symmetric
- link conditions are static
- the density of devices in an area is uniform
- the traffic demand and access patterns are fixed
- the communication pairs (i.e., source and destination devices) are fixed
- users move based on a random-walk model

In most of the cases, these assumptions are unrealistic and incorrect. For instance, it is known that, in general, the spatial distribution of network nodes moving according to the random waypoint model is nonuniform (e.g., [82]). Moreover, wireless channels can be *highly asymmetric* and *highly time-varying*. Unfortunately, there are not many traces of actual data access patterns or realistic models available for wireless users, especially for mobile peer-to-peer settings (e.g., [203]). Often academics are reluctant to expend the time and energy required to “sanitize” the data sets. Similarly, companies are not eager

to disclose information they consider proprietary. The development of realistic, but also general, tractable and elegant models is a non-trivial task.

In contrast to traditional wired-network topologies that reflect the physical hardwired connection of routers, wireless network topologies are more dynamic and have a stochastic element due to the radio propagation conditions, the user mobility, and client-AP association process. Modeling wireless network topologies opens up new research directions. For current traffic modeling tools, the application mixture and traffic models are quite simplistic. One of the problems is that complex mobility and topology models are rich sub-fields of their own expertise. There should be tools and methods for others to effectively and easily use models from these sub-fields in standard simulators. The scaling properties of simulators are very important and have not been fully addressed. For example, it is not clear that a simple 20-node simulation can be “stretched” to 10,000 node simulations by a “copy-and-paste” methodology.

A wide range of traffic load is observed in wireless campus-wide infrastructures. In general the traffic load is light, though there are long tails. Furthermore, APs in campus-wide infrastructures exhibit a dichotomy with respect to their upload and download traffic: there are APs dominated by uploaders and APs dominated by downloaders. The most popular applications are web browsing and peer-to-peer, accounting for approximately 81% of the total traffic, and most users are also dominated by these two applications.

Rich sets of empirical traces, collected from large-scale wireless infrastructures, impelled us to model the user and access demand, and thus, enable more meaningful performance analysis studies. We distinguished the following important dimensions in wireless network modeling:

- user demand
- access patterns
- network topology
- channel conditions

This distinction enabled us to superimpose models for the demand on a given topology and focus on the right level of detail. This monograph focused on user demand and access patterns, modeling session and flow parameters. Sessions capture the interaction between the clients and the network, while flows model the above-packet-level traffic activity masking the underlying network dependencies. The wireless access of a client is modeled as an alternation between sessions and disconnections. An access pattern is characterized by an arrival process at certain APs and a sequence of transitions between APs. Important parameters in access patterns are the arrival process at an AP, session duration, transitions between APs, and predictability of the next AP association.

In current campus-wide wireless infrastructures, the majority of the sessions last less than one hour. Wireless clients exhibited relatively low mobility, spending a large percentage of their wireless life at the same AP. In

general, mobile sessions tend to have a small percentage of long visits and a large percentage of short visits at APs. Markov-chain models can be used to characterize transitions of clients between APs and accurately predict the next AP with which a client will associate. These predictions can be further enhanced by incorporating networking and physical topological data as well as temporal information, such as time, day of the week, and visit duration. Time-varying Poisson processes can model client arrivals at APs well. Predicting client arrivals at APs can improve the buffering, caching, load balancing, and prefetching at APs in order to mask the end-to-end delay, particularly in the case of regular clients. APs may not only predict client arrivals but also traffic demand. Based on these predictions, neighboring APs can advise newly arrived clients to avoid hotspots, suggest alternative APs, and better balance their load and channel utilization.

Highlighting the ability of empirically-based models to capture the characteristics of the user workload and providing a flexible framework for using them in performance analysis studies was another contribution of this research. Specifically, a multi-level modeling of the wireless demand in IEEE802.11 campus-wide infrastructures was presented. A methodology for the statistical modeling of wireless network traffic demand was proposed relying on robust statistical methods to study large-scale phenomena. Furthermore, we contributed intuitive system-wide and AP-level models of traffic demand that capture the network-independent characteristics of the traffic workload. The parameters and the proposed statistical models appear in Table 6.1.

Parameter	Model
Session arrival	Time-varying Poisson
Client arrival	Time-varying Poisson
Flow inter-arrival/session	Lognormal
Flow number/session	BiPareto
Flow size	BiPareto
Session duration	BiPareto
Transitions between APs	Markov-chain

Table 6.1. Proposed models for wireless access and traffic demand.

The session- and flow-related models are well-behaved, robust, and reusable. We validated these models using different spatial scales (e.g., AP-level, network-wide, groups of APs located at the same building) and different periods and found that the same distributions apply for modeling at finer spatial scales. At each spatio-temporal scale, the models for sessions and flows remain the same with only their parameter values differing.

By selecting the appropriate spatio-temporal granularities of the models, the right balance between reusability and accuracy can be addressed. For example, when hourly periods and AP-scale are used, the models maintain suf-

ficient spatial detail at the cost of a lower scalability and amenability. When a network-wide scale is used, we gain simplicity at the cost of a higher loss of detail. The evaluation of the models was performed using statistics- and systems-based metrics. When the statistics-based metrics showed a deviation of the models from the empirical data, the systems-based metrics were used to evaluate the impact of this difference on the performance of that system. The systems-based evaluation focused on the performance of a hotspot AP, employing various metrics, such as the hourly aggregate throughput, per-flow delay and throughput, and goodput. We generated synthetic traces based on various models and spatio-temporal scales. Emulation- and simulation-based scenarios were performed using synthetic and original traces—generated from a real-life wireless infrastructure—as input for the user workload. The proposed models exhibit a performance which is very close to the one obtained when the original traces are used (“ground-truth” of the AP performance). On the other hand, naive models result in a performance that deviates substantially from the one reported when the original traces are used.

6.2 Directions for future research

Pervasive computing spaces involve *autonomous networked heterogeneous* systems operating with *minimum human* intervention. They should be capable of detecting impending violations of the service requirements, reconfiguring themselves, and isolating the failed or malicious components. To do this, it is necessary to provide *dynamic adaptation* mechanisms that perform the following tasks:

- monitor the environment
- relate low-level information about resource availability and network conditions to higher-level functional or performance specifications
- select the appropriate network interface, channel, AP, power transmission, and bitrate

Wireless networks exhibit vulnerabilities that can be classified into the following three main types:

- connectivity
- performance
- security

Connectivity problems reflect the lack of sufficient wireless coverage; an end-user may observe degraded performance—such as a low throughput or a high latency—due to various reasons related to the wired or wireless parts of the network, congestion in several networking components, or slow servers.

Security problems involve the presence of rogue APs and malicious clients. In mobile wireless networks, it is easier to disseminate worms, viruses, and

false information or eavesdrop, deploy rogue or malicious software or hardware, attack, or behave in a selfish or malicious manner. Attacks may occur at different layers, aiming to exhaust the resources, while instances of selfish behavior include promising falsely to relay packets or not responding to requests for service. Given the vulnerabilities of wireless networks, security provision needs to become a research target in its own right instead of being simply an add-on component, investigated in isolation to quality of service.

Our ultimate technological goal is to develop intelligent and robust wireless networks, which can be defined as networks of devices that adapt in a *self-organizing* and *autonomous* manner based on their resources to enhance their quality of service. Examples of important issues that need to be addressed are the following: efficient monitoring of networks, identifying the appropriate parameters to be measured that reflect accurately the network conditions, understanding the impact of these conditions on the performance of an application, and facilitating various mechanisms that enable wireless devices to select the appropriate network interface or channel.

6.2.1 Increasing capacity

To increase the network capacity, improvements in all protocol layers have been proposed. At the physical layer, advanced radio technologies, such as reconfigurable and frequency-agile radios, multi-channel and multi-radio systems, and directional and smart antennas have been proposed to increase capacity and mitigate impairments caused due to fading and co-channel interference. Multipath fading can be caused by phase cancellation between different propagation paths, reducing signal power against noise. These mechanisms need to be integrated with MAC and routing protocols.

Efficient spectrum utilization is an issue of primary importance. Studies have shown that there are frequency bands in the spectrum that are largely unoccupied most of the time while others are heavily used. Cognitive radios have been proposed to enable a device to access a spectrum band that is unoccupied by others at that location and time [265]. Cognitive radio is defined as an *intelligent wireless communication* system that is *aware of the environment and adapts to changes*, aiming to achieve both *reliable communication whenever needed* and *efficient utilization of the radio spectrum* [175, 265]. The commercialization of such technologies has not yet been fully realized, as most of them are still in research and development phases and face cost, complexity, and compatibility issues.

Other improvements target the MAC layer. To achieve a higher throughput and energy-efficient access, devices may use *multiple* channels instead of only one fixed channel [60]. Depending on the number of radios and transceivers, the following approaches can be distinguished:

- Single-radio MAC:

- Multi-channel single-transceiver MAC: one transceiver is available in the network device, and therefore only one channel is active at a time in each device.
- Multi-channel multi-transceiver MAC: the network device includes multiple RF front-end chips and baseband processing modules to support several simultaneous channels. A single MAC layer controls and coordinates the access to these multiple channels.
- Multi-radio MAC: the network device has multiple radios, each with its own MAC and physical layer.

Researchers have proposed modifications to the IEEE802.11 MAC to use multiple channels. These approaches can be classified into different categories depending on the channel assignment and availability of multiple transceivers. For example, one approach dedicates a channel to the control packets and uses the remaining channels for data packets, whereas another approach utilizes all channels identically. Two main trends appear when multiple transceivers are available: the multiple-transceivers with one transceiver per channel and the use of a common transceiver for all channels. Unlike the multi-transceiver case, a common transceiver operates on a single channel at any given point of time. Manufacturers, such as Engim and D-Link, have launched APs that use multiple channels simultaneously and claim to provide high-bandwidth wireless networks.

6.2.2 Capacity planning

Unlike device adaptation that takes place dynamically, capacity planning determines the AP placement, configuration, and administration of APs in an off-line proactive manner. The configuration of an AP includes the determination of its transmission power, frequency, and orientation. The determination of the transmission power is a trade-off between energy conservation and network connectivity. Reducing transmission power lowers the interference, which in turn, reduces the number of collisions and packet retransmissions. At the same time, it also results in a smaller number of communication links and lower connectivity. Another issue is related to the conservative configuration of the default carrier-sense threshold. An increase in the carrier-sense threshold of a device also results in an increase of the delay to transmit. A dynamic configuration of this threshold that takes into consideration the interference range of the potential receivers and the transmission power may enable a larger number of devices in proximity to transmit, improving the per-flow and aggregate throughput [346]. Capacity planning aims to provide sufficient coverage and satisfy demand, considering the spatio-temporal evolution of the demand. Typical objectives include:

- the minimization of interference
- the maximization of the coverage area and overall signal quality
- the minimization of the number of APs used to provide sufficient coverage

Capacity planning is an important research direction and has been the focus of several research efforts (e.g., [240, 309, 213, 306, 59]). Several capacity planning systems assume predefined positions of the APs and aim to reduce the number of APs used based on administrative criteria. Power management—an integral component of capacity planning—aims to control spectrum spatial reuse, connectivity, and interference. An objective of power control could be to adjust the transmit power of devices, such that their signal-to-interference-noise-ratio (SINR) meets a certain threshold required for an acceptable performance (e.g., [247, 251, 250, 249, 96, 269, 154, 225]). The non-deterministic nature of the environment due to exogenous parameters, mobility, and radio propagation characteristics impact the performance of the network, making capacity planning challenging and further motivating the need of dynamic network adaptation.

6.2.3 Network interface and channel selection

The problem of channel assignment has been studied in the context of cellular networks. The spectrum is divided into a number of non-interfering disjoint channels using different techniques, such as:

- frequency division, in which the spectrum is divided into disjoint frequency bands
- time division, in which the channel usage is allocated into time slots
- code division, in which different users are modulated by spreading codes
- space division, in which users can access the channel at the same time and the same frequency by exploiting the spatial separation of the individual user. Multibeam (directional) antennas are used to separate radio signals by pointing them along different directions

The channel or network interface selection can be static or dynamic. The decision of which channel or network interface to select can be based on various criteria, such as the AP capacity, channel quality, application requirements, registration cost, and admission control. In current infrastructure networks, a common criterion for selecting an AP is based on received signal-strength values, which indicate the quality of the wireless link of a client to an AP and affect the client transmission rate. Although signal-strength does impact the packet delivery probability, signal-strength measurements is not an optimal metric for AP selection due to the asymmetry and highly time-varying characteristics of link conditions. Other criteria combine link quality and traffic-load estimations, including the number of active clients, average amount of time an AP spends to serve its users, beacon delays, packet error rate, and round-trip-time estimations (e.g., [348, 336]). Note that both the traffic load and link conditions can impact these parameters, so it is important to collect sufficient measurements in appropriate temporal scales and layers to obtain a clear picture of the network conditions. IEEE802.11k is a proposed standard

that focuses on how a wireless LAN should perform channel selection and power control to optimize its performance.

Typical IEEE802.11b devices reduce their bit-rate when repeated unsuccessful frame transmissions are detected. Furthermore, their performance is considerably degraded in the presence of a host with a reduced bit-rate. In general, in a wireless infrastructure, the client's bit-rate, use of the RTS-CTS mechanism, and frame size can impact its performance. Rate adaptation enables wireless devices to select the best transmission rate and dynamically adapt their decision to the time-varying channel quality. Typical metrics for estimating the channel quality include the signal-to-noise ratio (SNR) and the delivery probability of probing packets.¹ Various bit-rate adaptation mechanisms have been proposed in the literature (e.g., [214, 238, 186, 325, 305, 359, 172, 173, 86, 304, 222, 59]).

APs in proximity, configured in the same or overlapping channels may interfere with each other, affecting dramatically the user performance. To alleviate the interference, APs and clients may dynamically switch channels or adapt their transmission power. Channel selection algorithms need to address several issues, such as

- fast discovery of devices across channels
- fairness across active flows and participants
- accurate measurements of varying channel conditions

Several studies on channel switching mechanisms have appeared recently, e.g., [121, 330, 81, 143, 273, 246, 200, 197, 111, 235]. Rate adaptation and channel and network interface selection face a fundamental challenge: in order to be effective they require an accurate estimation on-the-fly of channel conditions in the presence of various dynamics caused by fading, mobility, and hidden terminals. This involves distributed and collaborative monitoring and analysis of the collected measurements. Their realization in an energy-efficient manner is a non-trivial task.

6.2.4 Monitoring

Depending on the type of conditions that need to be measured, monitoring needs to be performed at certain layers and spatio-temporal granularities. Monitoring tools are not without flaws and several issues arise when they are used in parallel for *thousands* devices of *different* types and manufacturers. These issues are related to:

- fine-grain data sampling

¹ Rate adaptation, a link-layer mechanism, is left unspecified by IEEE802.11 standards. The current specification mandates multiple transmission rates at the physical layer that use different modulation and coding schemes. For example, IEEE802.11b supports four transmission rates (1-11 Mbps), IEEE802.11a eight rates (6-54 Mbps), and IEEE802.11g twelve (1-54 Mbps).

- time synchronization
- incomplete information
- data consistency

Often monitoring tools are limited in their capabilities because they cannot capture all the relevant information due to either hardware limitations, the proprietary nature of hardware and software, or hidden terminals. Furthermore, there are many protocol features of IEEE802.11, such as those related to the rate adaptation and transmission power control, whose implementations are vendor-specific and whose details are not publicly available.

Extensive monitoring and collection of data in fine spatio-temporal detail can improve the accuracy of the performance estimates, but also increase the *energy consumption* and *detection delay*, as the network interfaces need to monitor the channel over longer time periods and then exchange this information with other devices. Four important aspects that need to be addressed are:

- identification of the dominant parameters through sensitivity analysis studies
- strategic placement of monitors at routers, APs, clients, and other devices
- automation of the monitoring process to reduce human intervention in managing the monitors and collecting data
- aggregation of data collected from distributed monitors to improve the accuracy, while maintaining a low communication and energy overhead

To provide a more complete picture of the network conditions, cross-layer measurements—collected data spanning from the physical layer up to the application layer—are required. This further complicates the monitoring and analysis process. To interpret their dependencies and identify the relevant explanatory variables, cross-correlation functions on this data can be employed in an iterative approach. The wireless domain gives many opportunities for the use of a rich set of statistical and visualization techniques, such as feature extraction, multidimensional clustering, and forecasting. Also, the identification of the impact of various benchmarks is critical for the support of intelligent and robust wireless networks.

Benchmarks can be derived from theoretical models or by analyzing real-life data or by combining in different *temporal*, *spatial*, and *network* scales. They may also reflect different *perspectives* (e.g., user, client, AP, group of APs in certain regions, entire infrastructure). In general, the availability of benchmarks can play a dramatic role in comparative performance analysis and validation studies through repeatability. Examples of such benchmarks can be combinations of the following non-exhaustive general metrics:

- application characteristics
- device mobility
- robustness and fault-tolerance criteria
- network conditions

- network topologies

An application can be characterized based on its requirements (e.g., in terms of throughput, delay, jitter, packet losses, resolution, media quality, and “user-satisfaction” metrics), interactivity model, usage pattern, and traffic demand. Depending on the environment, the device mobility could be

- group or individual
- spontaneous or controlled
- pedestrian or vehicular
- known *a priori* or dynamic

Examples of robustness and fault-tolerance criteria include the number of active neighboring devices, the degree of vulnerability under the loss of valuable links or APs, and the impact of induced failures on the performance. Network conditions can be characterized by link quality criteria (e.g., packet losses, delays, signal-to-noise ratio), the spatio-temporal distributions of traffic demand and application mix, and the distributions of regions of weak connectivity or no signal. Network topologies can be described based on their connectivity and link characteristics, distribution and density of peers, degree of clustering, co-residency time, inter-contact time, duration of disconnection from the Internet, and interaction patterns.

6.3 Bio-inspired computing networks

Computing spaces with wirelessly-enabled devices monitoring the environment, and processing and communicating the acquired information are becoming more and more pervasive. In several situations, specialized devices communicate with each other to aggregate their information and deliver it to the user in the appropriate modality and format. In other situations, miniature networked devices need to collaborate and form a network that presents intelligent and robust behavior.

Depending on the degrees of collaboration, caching and network paradigms, wirelessly-enabled devices in pervasive computing spaces interact, sharing information and other resources. In this resource sharing, better routes, APs, servers, and caches can be selected, based on various criteria, such as: energy-efficiency, response delay, throughput, network lifetime, robustness, fault-tolerance, security, scalability, and user interruption. Similarly, devices compete for, or allocate resources to optimize these criteria.

Several social systems in nature, composed of simple individuals exhibit an intelligent collective behavior. Researchers have been drawing parallels between biological and computer systems and applying biologically-inspired models to achieve more efficient computing paradigms. In a particularly interesting work, Weitz *et al.* [353] draw several analogies among different disciplines that study the evolution of networks; mathematicians and physicists

focus on the network structure as it changes over time, while biologists investigate how selection and fitness act to optimize the performance of a biological network. Examples of biological networks are the metabolic, regulatory, and protein networks. Like biologists, computer scientists seek to apply energy-efficient adaptation mechanisms to optimize networking environments.

Biologists have been studying the structure and behavior of organisms in depth, such as the *C. elegans*, which is the first multicellular animal to have a fully-sequenced genome and a major model organism used for biomedical research. Two interesting questions about biological networks are the following:

- Do biological networks reflect hidden organizational and structural principles?
- How do these principles contribute to the adaptation, fault-tolerance, and energy-efficiency of the biological organisms?

Some researchers have suggested that biological networks are organized through scale-free random evolution while others have claimed that they exhibit statistically significant patterns. Watts and Strogatz showed that metabolic and *C. elegans* networks have a high degree of clustering and a short average length. Barabasi studied the network of protein interactions in yeast and found that the most highly-connected proteins are the most important for the survival of a cell. Scale-free networks have been found to be resistant to random failures but vulnerable to attacks against their “key” nodes (e.g., hubs, nodes with high degree of connectivity). Could we build more adaptive, robust and energy-efficient pervasive computing systems by applying the analogies drawn from these structural properties of biological networks?

Diffusion has been studied in biology and similarities can be drawn between the propagation of pathogens, such as viruses and worms, or other type of information in computer networks, and the proliferation of pathogens in cellular organisms [160]. Chemotaxis is the kind of taxis in which cells, bacteria, and other single-cell or multicellular organisms direct their movements according to certain chemicals in their environment, critical to their development and normal function. In different spatio-temporal scales, the information dissemination in pervasive computing environments plays a similar role. Could we improve the information dissemination in pervasive computing environments by applying chemotaxis-inspired mechanisms?

Other examples are sensory networks in which biologists explore strategies used by cells to function reliably under the presence of noise. Similarly, routing algorithms have been inspired by models related to ant colonies and the notion of stigmergy, that is the indirect communication in a self-organizing emergent system where its individuals communicate with one another through modifications induced in their local environment. Real ants have been shown to find intelligent solutions to problems, such as discovering shortest paths using only the pheromone trail deposited by other ants, prioritizing food sources based on their distance and ease of access, carrying large items, and forming bridges.

Positioning and orientation is another area for cross-disciplinary research. Let us take as an example birds and their ability to navigate and orient themselves when displaced. This ability is a complex phenomenon, which may include both endogenous programs as well as learning. Studies have shown that birds use several mechanisms, such as landmarks, solar cues (“sun compass”), stellar cues, and geomagnetic cues. There is also some evidence that odors and sounds may provide additional cues. More recent research has found a neural connection between the eye and “Cluster N”, the part of the forebrain that is active during migrational orientation, suggesting that birds may actually be able to see the magnetic field of the earth. The transfer of knowledge is realized in both directions; for example, radiotelemetry and sensor networks have been used extensively in ornithology and marine biology to monitor animals and their habitat. Efficient algorithms for searching, managing, visualizing, data mining, and pattern matching have been also applied to biological data. Cross-disciplinary research is emerging to explore how computer scientists can use properties from biological systems in building efficient pervasive computing spaces and how biologists can experiment with simulations from large-scale computer networks to better understand their own biological networks.

6.4 New horizons in cross-disciplinary research

Computer science has offered new paradigms, technologies and tools for communication and interaction that were catalysts not only in other sciences but also in society. On-line collaboration has been enriched with new applications and tools for storing, sharing, and experimenting with multimedia data. Mobile peer-to-peer computing may enhance the formation of on-line communities of mobile users and create new socio-technological paradigms. In a recent study, the World Bank computed the time elapsed between the invention of various technologies across last centuries and their widespread adoption. While telephones reached 80% country coverage in 100 years, radio in 65 years, and Internet use in 22 years, mobile phones required just 16 years. It remains to be seen if the mobile peer-to-peer paradigm will trigger the formation of new on-line communities and have a greater social penetration.

According to the Internet World Stats 2007, Internet penetration in North America is 69.7% of population compared to 3.6% for Africa and 10.7% for Asia. There are already several discussions, research proposals, market initiatives, and political actions on communication technologies and infrastructures for developing regions. Wireless networking and mobile peer-to-peer computing would be two candidates for bridging the digital divide.

The mobile peer-to-peer paradigm with its distinct feature of cooperation can be applied to facilitate the information access and sharing among devices for the support of context-aware services. An underlying objective of these services is the recognition and characterization of the users’ context without interrupting them from their main tasks. This involves research in domains that

span from networking and systems to contextual information representation and reasoning, and graphics. Thus mobile peer-to-peer computing, combined with context-aware computing, opens up exciting challenges in computer science, demanding interdisciplinary research and innovative paradigms.

The new technologies and rapid growth and distribution of data impose new ethical and social questions encompassing issues spanning from privacy and security to medical and legal considerations. To highlight the variety of new issues involved in mobile access, let us focus on a specific topic: mobile electronic identity. Wirelessly-enabled devices that would support such mobile electronic identification mechanisms are vulnerable to different types of threats, such as impersonation, eavesdropping on personal data, and dissemination of false information, viruses, and spam. These vulnerabilities and constraints make the provisioning of privacy, confidentiality, and security a challenging task. Researchers predict that electronic tags—such as RFTags—will be pervasive; not only as electronic identification but also as implantable chips in humans, raising even more questions about security, privacy, confidentiality, legislation and ethics. Not only technological and legislative, but also environmental issues arise. Several environmental reports call attention to the hazardous materials used in the phones and batteries including arsenic, antimony, beryllium, cadmium, and lead. Disposing such materials into our soil and water creates an enormous amount of toxic garbage, demanding urgently efficient recycling programs. It is the responsibility of our community to raise relevant questions and encourage the investigation of those issues. Science-fiction scenarios speculate about the “seventh sense”, the technologically-enhanced ability of humans to observe and understand the environment. This crossing of mobile computing, wireless technologies, and multi-modal interfaces (e.g., tactile and haptic displays, tagging and sensing technologies) creates even more networking paradigms. Extended by augmented reality and brain/user-interface technologies, this interdisciplinary research creates new fertile realms in education, medicine, entertainment, assistive technology, psychology, law, art, ethics, and urban design. The deployment of wireless and augmented reality technologies will raise new issues and challenges in how to create environments for maximum human development that can guarantee everybody the best possible development under conditions of freedom and safety. Pervasive computing spaces intertwined with urban environments should not prevent people from developing a harmonious contact with others and with nature.

Wireless technology—and in general, computer science—is playing a dramatic role in our lives not only by assisting other sciences, but by reshaping society and the way we think and sense.

A

Appendix

Model	PDF
Normal	$p(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-(x-\mu)^2/(2\sigma^2)}, x \in (-\infty, \infty)$
Lognormal	$p(x) = \frac{1}{x\sigma\sqrt{2\pi}}e^{-(\ln x - \mu)^2/(2\sigma^2)}, x \in (0, \infty)$
Exponential	$p(x) = \frac{1}{\mu}e^{-x/\mu}, x \in [0, \infty)$
Gamma	$p(x) = \frac{1}{b^\alpha \Gamma(\alpha)}x^{\alpha-1}e^{-x/b}, x \in [0, \infty)$
Rayleigh	$p(x) = \frac{x}{b^2}e^{-x^2/(2b^2)}, x \in [0, \infty)$
Weibull	$p(x) = b\alpha^{-b}x^{b-1}e^{-(x/\alpha)^b}, x \in [0, \infty)$
Generalized Gaussian	$p(x) = \frac{b}{2\alpha \Gamma(1/b)}e^{-(x /\alpha)^b}, x \in (-\infty, \infty)$
Pareto	$p(x; k, x_m) = k\frac{x_m^k}{x^{k+1}}, x \geq x_m$
BiPareto	$p(x) = k^b(1+c)^{b-\alpha}x^{-\alpha-1}(x+kc)^{\alpha-b-1}(bx+\alpha kc), x > \alpha$

Table A.1. Models used in demand analysis.

B

Wireless measurement-based data repositories

Measurement-based data collected from diverse wireless networking environments, such as metropolitan areas, vehicular, houses, academic environments, research labs, and conference sites, have been made available in various data repositories. One of the largest collections with publicly available wireless traces is CRAWDAD [10] which hosts traces from different wireless environments. Tables B.1, B.2, and B.3 summarize the type of wireless traces available in CRAWDAD. Other archives include:

- the UCSD wireless topology discovery trace [56, 260]
- the MIT Roofnet [43]
- the MobiLib [30]
- wireless LAN traces from the ACM Sigcomm'01 [55]
- vehicular network traces [99, 153, 156, 95]

Empirical studies focusing on metropolitan area-based wireless networks have recently taken place:

- in Cambridge, UK with users currying iMotes [241]
- in Toronto with Bluetooth-enabled PDA users walking in the subway and malls to test if a worm outbreak is possible in practice [335]
- at MIT, with one hundred smart phones that use both short-range (such as Bluetooth) and long-range (GSM) networks logging users' location, communication, and device usage behavior information [133]
- in Cambridge, US, with users of Roofnet, an experimental IEEE802.11b/g mesh network which provides broadband Internet access, developed at MIT CSAIL [43]
- in a grid of six nodes placed within three different houses that produced wireless measurements to characterize connectivity and UDP and TCP throughput [291, 362]
- in Oulu, Finland, panOULU network provides in its coverage area wireless broadband Internet access in libraries, schools, sports facilities, hospitals, and the market area [40]

The Roofnet measurements focused on the link-level of IEEE802.11, finding high-throughput routes in the face of lossy links, adaptive bit-rate selection, and developing new protocols which take advantage of wireless communications' unique properties [85, 87].

Empirical measurement studies were also performed in several conferences, such as: the 2005 IETF meeting [205], 2004 ACM Sigcomm[317], and 2001 ACM Sigcomm[55], in which SNMP, and SYSLOG traces were acquired from the deployed IEEE802.11 APs.

Traces from large-scale academic deployments of IEEE802.11 APs include UNC [51], Dartmouth [10], USC [30], and smaller-scale IEEE802.11 APs networks in research labs or institutes, such as FORTH [51], and IBM [75, 76]. Sensor-based testbeds include the one at Columbia University using TinyOS on Mica2 motes for testing a MAC protocol [135, 136].

Vehicular-based networking environments have also been explored [153, 95]. For example, [153] includes traces from a short-range communications between vehicle and roadside traffic and [95], from an IEEE802.11-enabled bus in a campus and the surrounding county in UMASS. Traces from a CDMA 1x EV-DO network are also available [152]. Finally, a very large collection of data tailored to measuring Internet traffic and performance can be found at the CAIDA site [11].

Table B.1. IEEE802.11 Infrastructure Network					
Traces	Scale		Type	Time period	Time granularity
	# Devices	# APs			
UNC	9881	574	SYSLOG, SNMP, TCP & UDP headers, HTTP requests	29/9/2004-30/11/2004 29/9/2004-26/6/2005 13/4/2005-20/4/2005	SNMP:300 s
FORTH	1	12	signal strength	12/12/2007	SNMP:300 s signal strength:150 s
FORTH & Crete Aquarium	1	8	signal strength	15/11/2007-16/11/2007	SNMP:300 s signal strength:150 s
Dartmouth	2500	566	SYSLOG, TCP & UDP headers	11/4/2001-4/10/2005	SNMP:300 s
ACM Sigcomm 2001	195	4	TCP & UDP headers	27/8/2001-31/8/2001	SNMP:60 s
IBM/Watson	1366	177	–	20/7/2002-18/8/2002	SNMP:300 s
Stanford/Gates	74	12	SYSLOG, TCP & UDP headers	20/9/1999-12/12/1999	SNMP:120 s

Table B.2. IEEE802.11 Ad-hoc Network (Controlled experiments, mostly on routing)

Traces	Scale # Devices # APs	Type	Time period	Sampling	User position	Mobility	Area
Dartmouth/outdoor	33	N/A	UDP,IP&MAC headers, 17/10/2003	6 or 10 s	GPS	vehicular	outdoors (Athletic field)
Rutgers/noise	64	N/A	MAC frames, signal strength 15/10/2005	30 s	predefined	stationary	indoors (ORBIT wireless testbed)
UCSB/meshnet	20	N/A	expected transmission time 1/4/2006-7/4/2006	10 s	predefined	stationary	indoors (UCSB Meshnet)

Table: B.3. Ad-hoc Network Type, IEEE802.11 wireless technology (Controlled experiments, mostly on throughput & connectivity)

Traces	Scale # Devices # APs	Type	Time period	Sampling	User position	Mobility	Tool	Area
Intel	6	N/A	TCP & UDP 29/6/2005-3/7/2006 throughput data	60 s	predefined	stationary	netperf	indoors (Houses)
Gatech	2	N/A	GPS, bytes received, signal strength 24/1/2005-23/4/2005	1 s	GPS	vehicular	iperf	outdoors (Highway)
Synusb/Mobisteer	1	6	MAC frames, GPS 28/9/2006-29/11/2006	1 s	GPS	vehicular	N/A	outdoors (Parking lot, apartment complex)
Microsoft	2	11	application level, MAC frames, GPS 22/1/2001-16/1/2001	1 s	GPS	vehicular	N/A	outdoors (Microsoft campus)

References

1. (2007) Wi-Fi chipsets shipments reach 200 million in 2006-report. http://www.telecomseurope.net/article.php?id_article=3549/.
2. Ajax. <http://en.wikipedia.org/wiki/AJAX>.
3. Ajax: A new approach to web applications. <http://www.adaptivepath.com/publications/essays/archives/000385.php>.
4. AMBIENTE, Division at Fraunhofer IPSI, Darmstadt, Germany. <http://www.ipsi.fhg.de/ambiente>.
5. America's Most Connected Campuses. <http://forbes.com/home/lists/2004/10/20/04conncampland.html>.
6. Aura at Carnegie Mellon. <http://www.cs.cmu.edu/~aura/>.
7. Barnsley telehealth service monitors heart failure patients in the home. <http://www.mtbeurope.info/news/2006/603025.htm>.
8. The bat ultrasonic location system. <http://www.cl.cam.ac.uk/research/dtg/attachive/bat/>.
9. Bluetooth biosensing wristwatch monitors heart rate, activity and emotions. <http://www.mtbeurope.info/news/2006/607027.htm>.
10. CRAWDAD a community resource for archiving wireless data at Dartmouth. <http://crawdad.cs.dartmouth.edu/>.
11. Data collection at CAIDA. <http://www.caida.org/data/>.
12. Delay tolerant networking research group. <http://www.dtnrg.org/>.
13. Ekahau v.3.1. (<http://www.ekahau.com>).
14. eTForecasts report on worldwide PDA markets. <http://www.etforecasts.com/pr/pr0603.htm>.
15. Ethical Implications of Emerging Technologies: A Survey. Prepared by Mary Rundle and Chris Conley. United Nations Educational, Scientific and Cultural Organization. <http://unesdoc.unesco.org/images/0014/001499/149992E.pdf>.
16. Free real-time traffic maps, alerts, and Jam Factor reports for the routes you drive. <http://www.traffic.com/>.
17. Free riding on gnutella. <http://www.firstmonday.org/issues/issue5.10/adar/>.
18. Fuego-Helsinki Institute for Information Technology Future Mobile and Ubiquitous Computing Research Program. <http://www.hiit.fi/fuego>.
19. Future Computing Environment Group. <http://www.cc.gatech.edu/fce/index.html>.
20. Future computing environments. <http://www.cc.gatech.edu/fce/smartfloor/>.

21. GE to distribute MP4 remote foetal and maternal monitoring system for PDA. <http://www.mtbeurope.info/news/2006/602010.htm>.
22. Group of User Interface Research. <http://guir.berkeley.edu>.
23. IBM Pervasive Computing. <http://www-306.ibm.com/software/info1/websphere/index.jsp?tab=products/-mobilespeech>.
24. LimeWire homepage. <http://www.limewire.org>.
25. Ling Liu. Mobile web and location-based services: Opportunities and challenges. <http://www.cc.gatech.edu/~lingliu/keynotes/>.
26. Ling Liu. Security and trust in peer-to-peer systems: Risks and countermeasures. <http://www.cc.gatech.edu/~lingliu/keynotes/>.
27. Location-based services. <http://www.lbsinsight.com>.
28. Medical Technology Business Europe: Patient Monitoring. <http://www.mtbeurope.info/patientmonitoring/index.htm>.
29. Microsoft EasyLiving. <http://research.microsoft.com/easyliving/>.
30. Mobilib: Community-wide library of mobility and wireless networks measurements. <http://nile.usc.edu/MobiLib/>.
31. Movies on the Move - First GPS-enabled Movie Guide in the U.S. <http://www.lbsinsight.com/?id=658>.
32. NAVTEQ, digital map data. <http://www.navteq.com/>.
33. Null hypothesis, from wikipedia. (http://en.wikipedia.org/wiki/Null_hypothesis).
34. NYC wireless. <http://www.nycwireless.net>.
35. Onstar. <http://www.onstar.com>.
36. OpenFT is a file sharing protocol developed by the giFT project. <http://en.wikipedia.org/wiki/OpenFT>.
37. Peer-to-peer: Harnessing the power of disruptive technologies. <http://www.freehaven.net/doc/oreilly/accountability-ch16.html>.
38. Portolano: An expedition into invisible computing. <http://portolano.cs.washington.edu/>.
39. Precise Real-time Location. <http://www.ubisense.net/>.
40. Public access network oulu. <http://www.panoulu.net/>.
41. RANDOM GRAPHS and COMPLEX NETWORKS (class offered by David Aldous). <http://www.stat.berkeley.edu/users/aldous/Networks/>.
42. Research Programme on Proactive Computing. <http://www.aka.fi/index.asp?id=d16162a4696042d2821b6291f0732a8e>.
43. Roofnet is an experimental 802.11b/g mesh network in development at MIT. <http://pdos.csail.mit.edu/roofnet/doku.php>.
44. RSA Laboratories. <http://www.rsa.com/rsalabs/node.asp?id=2002>.
45. Self-organizing neighborhood wireless mesh networks. <http://research.microsoft.com/mesh/>.
46. Silicon architectures for wireless systems by Prof. Rabaey. <http://bwrc.eecs.berkeley.edu/People/Faculty/jan/presentations/Lausanne/-lecture4.pdf>.
47. Smart Spaces at NIST. <http://www.nist.gov/smartspace/>.
48. Sony Computer Science Labs. <http://www.csl.sony.co.jp/>.
49. The Freenet project. <http://freenetproject.org/>.
50. Ubiquitous Computing at TECO. <http://ubicomp.teco.edu/index2.html>.
51. UNC/FORTH archive of wireless traces, models, and tools. <http://www.ics.forth.gr/mobile/>.

52. Verizon Wireless. <http://www.verizonwireless.com>.
53. Wavemarket location intelligence. <http://www.wavemarket.com/>.
54. Wireless and mobility extensions to ns-2. <http://www.monarch.cs.cmu.edu/cmu-ns.html>.
55. Wireless LAN traces from ACM SIGCOMM'01. <http://sysnet.ucsd.edu/pawn/sigcomm-trace/>.
56. Wireless topology discovery at ucsd. <http://sysnet.ucsd.edu/wtd/>.
57. Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b Mesh network. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, Portland, OR, USA, August 2004.
58. Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self-management in chaotic wireless deployments. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 185–199, Cologne, Germany, August 2005.
59. Aditya Akella, Glenn Judd, Srinivasan Seshan, and Peter Steenkiste. Self management in chaotic wireless deployments. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Cologne, Germany, August 2005.
60. Ian Akyildiz and Xudong Wang. A survey on wireless mesh networks. *IEEE Radio Communications*, 43(9):S23–S30, September 2005.
61. Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks. arXiv report cond-mat/0106096, June 2001.
62. F. Anjum, M. Elaoud, D. Famolari, A. Ghosh, R. Vaidyanathan, A. Dutta, P. Agrawa, T. Kodama, and Y. Katsube. Voice performance in WLAN networks-an experimental study. In *IEEE Conference on Global Communications (GLOBECOM)*, San Francisco, December 2003.
63. I. Antoniou, V. V. Ivanov, Valery V. Ivanov, and P. V. Zrelov. Principal Component Analysis of Network Traffic Measurements: the “Caterpillar”-SSA approach. In *Int. Workshop on Advanced Computing and Analysis Techniques in Physics Research, ACAT'2002*, Moscow, Russia, June 2002.
64. Somil Asthana and Dimitris Kalofonos. The problem of bluetooth pollution and accelerating connectivity in bluetooth ad-hoc networks. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (Percom)*, New York, NY, USA, March 2005.
65. R. Atkinson. IP encapsulating security payload. RFC 1827, August 1995.
66. R. Atkinson. Security architecture for the Internet protocol. RFC 1825, August 1995.
67. A. Auvinen, M. Vapa, M. Weber, N. Kotilainen, and J. Vuori. Chedar: Peer-to-peer middleware. In *Proceedings of the 19th IEEE International Parallel & Distributed Processing Symposium (IPDPS 2006)*, Rhodes Island, Greece, July 2006.
68. François Baccelli, Sridhar Machiraju, Darryl Veitch, and Jean Bolot. The role of PASTA in network measurement. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, pages 231–242, Pisa, Italy, September 2006.
69. Adam Back. Hashcash - a denial of service counter-measure. Technical report, <http://www.cypherspace.org/~adam/hashcash/>, August 2002.

70. P. Bahl, R. Chandra, and J. Dunagan. Ssch: Slotted seeded channel hopping for capacity improvement in IEEE802.11 ad-hoc wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, USA, September 2004.
71. Paramvir Bahl and Venkata Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *IEEE Conference on Computer Communications (InfoCom)*, Tel Aviv, Israel, March 2000.
72. Paramvir Bahl, Venkata N. Padmanabhan, and Anand Balachandran. Enhancements to the radar user location and tracking system. Technical report, Microsoft Research, Redmond, WA, February 2000.
73. Norman T. Bailey. *The mathematical theory of infectious diseases and its applications*. Hafner, 1975.
74. Anand Balachandran, Geoffrey Voelker, Paramvir Bahl, and Venkat Rangan. Characterizing user behavior and network performance in a public wireless lan. In *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, California, CA, USA, June 2002.
75. Magdalena Balazinska and Paul Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *First International Conference on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, USA, May 2003.
76. Magdalena Balazinska and Paul Castro. CRAWDAD data set ibm/watson (v. 2003-02-19). Downloaded from <http://crawdad.cs.dartmouth.edu/ibm/watson>, February 2003.
77. Udana Bandara, Mikio Hasegawa, Masugi Inoue, Hiroyuki Morikawa, and Tomonori Aoyama. Design and implementation of a bluetooth signal strength based location sensing system. In *IEEE Radio and Wireless Conference (RAWCON)*, Atlanta, GA, USA, September 2004.
78. Nikhil Bansal and Zhen Liu. Capacity, delay and mobility in wireless ad-hoc networks. In *IEEE Conference on Computer Communications (InfoCom)*, San Francisco, California, September 2003.
79. Daniel Barbara and Tomasz Imielinski. Sleepers and workaholics: Caching strategies in mobile environments. In *ACM SIGMOD International Conference on Management of Data*, pages 1–12, Minneapolis, Minnesota, June 1994.
80. Paul Barford and Mark E. Crovella. Generating representative Web workloads for network and server performance evaluation. In *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 151–160, Madison, Wisconsin, June 1998.
81. Mohammed Benaissa, Vincent Lecuire, F. Leage, and A. Shaff. Analysing end-to-end packet delay and loss in mobile ad hoc networks for interactive audio applications. In *Workshop on Mobile Ad Hoc Networking and Computing*, pages 27–33, Sophia-Antipolis, France, March 2003.
82. Christian Bettstetter, Giovanni Resta, and Paolo Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(3):257–269, July 2003.
83. Amiya Bhattacharya and Sajal K. Das. LeZi-update: an information-theoretic approach to track mobile users in PCS networks. In *Proceedings of the Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 1–12, Seattle, Washington, USA, August 1999.

84. Giuseppe Bianchi, Antonio Di Stefano, Costantino Giaconia, Luca Scalia, Giovanni Terrazzino, and Ilenia Tinnirello. Experimental assessment of the backoff behavior of commercial IEEE802.11b network cards. In *IEEE Conference on Computer Communications (InfoCom)*, pages 1181–1189, Anchorage, Alaska, USA, May 2007.
85. John Bicket, Daniel Aguayo, Sanjit Biswas, and Robert Morris. Architecture and evaluation of an unplanned 802.11b mesh network. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Cologne, Germany, August 2005.
86. John C. Bicket. Bit-rate selection in wireless networks. Master's thesis, Massachusetts Institute of Technology, February 2005.
87. Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, Philadelphia, PA, August 2005.
88. Matt Blaze, John Ioannidis, and Angelos D. Keromytis. Offline micropayments without trusted hardware. In *Proceedings of Financial Cryptography*, Cayman Islands, British West Indies, February 2001.
89. Rajendra V. Boppana and Satyadeva P. Konduru. An adaptive distance vector routing algorithm for mobile, ad hoc networks. In *IEEE Conference on Computer Communications (InfoCom)*, Anchorage, Alaska, April 2001.
90. Sem C. Borst and Nidhi Hegde. Integration of streaming and elastic traffic in wireless networks. In *INFOCOM*, pages 1884–1892, Anchorage, Alaska, USA, May 2007.
91. Lee Breslau, Pei Cao, Li Fan, Graham Phillips, and Scott Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *IEEE Conference on Computer Communications (InfoCom)*, New York, NY, March 1999.
92. Josh Broch, David Maltz, David Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, October 1998.
93. Ioannis Broustis, Konstantina Papagiannaki, Srikanth V. Krishnamurthy, Michalis Faloutsos, and Vivek Mhatre. MDG: measurement-driven guidelines for 802.11 WLAN design. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 254–265, Montreal, Quebec, Canada, September 2007.
94. Raffaele Bruno and Franca Delmastro. Design and analysis of a bluetooth-based indoor localization system. Technical report, Institute for Informatics and Telematics, Pisa, Italy, 1999.
95. J. Burgess and B. N. Levine. CRAWDAD data set umass/diesel (v. 2006-01-17). Downloaded from <http://crawdad.cs.dartmouth.edu/umass/diesel>, January 2006.
96. Martin Burkhart, Pascal von Rickenbach, Roger Wattenhofer, and Aaron Zollinger. Does topology control reduce interference? In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Roppongi Hills, Tokyo, Japan, May 2004.
97. Levente Buttyan and Jean-Pierre Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad-hoc networks. Technical Report DSC/2001/001, Swiss Federal Institute of Technology, Lausanne, January 2001.
98. Levente Buttyan and Jean-Pierre Hubaux. *Security and Cooperation in Wireless Networks*. Cambridge University Press, November 2007.

99. Vladimir Bychkovsky, Bret Hull, Allen K. Miu, Hari Balakrishnan, and Samuel Madden. A measurement study of vehicular internet access using in situ Wi-Fi networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Los Angeles, CA, September 2006.
100. Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, September 2002.
101. Juan-Carlos Cano and Pietro Manzoni. A performance comparison of energy consumption for mobile ad-hoc network routing protocols. San Francisco, California, USA, August 2000.
102. Jin Cao, William S. Cleveland, Dong Lin, and Don X. Sun. On the nonstationarity of Internet traffic. In *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 102–112, Cambridge, MA, USA, June 2001.
103. Yinhe Cao, Wen wen Tung, Jiafeng Gao, Vladimir A. Protopopescu, and Lee M. Hively. Detecting dynamical changes in time series using the permutation entropy. *Physical Review*, E70(4), October 2004.
104. Srdjan Capkun, Maher Hamdi, and Jean-Pierre Hubaux. GPS-Free Positioning in Mobile Ad-Hoc Networks. In *Proceedings of Hawaii International Conference On System Sciences*, Hawaii, January 2001.
105. George Casella and Roger L. Berger. *Statistical Inference, Second edition*. Duxbury, June 2001.
106. Paul Castro, Benjamin Greenstein, Richard Muntz, Parviz Kermani, Chatschik Bisdikian, and Maria Papadopouli. Locating application data across service discovery domains. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 28–42, Rome, Italy, July 2001.
107. Ramon Caceres, Peter Danzig, Sugih Jamin, and Danny Mitel. Characteristics of wide-area TCP/IP conversations. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, Zurich, Switzerland, September 1991. ACM.
108. Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on the design of opportunistic forwarding algorithms. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
109. Chris Chambers, Wuchang Feng, Sambit Sahu, and Debanjan Saha. Measurement-based characterization of a collection of on-line games. In *ACM Internet Measurement Conference (Sigcomm)*, Philadelphia, PA, USA, August 2005.
110. Probal Chaudhuri and J. S. Marron. Sizer for exploration of structures in curves. *Journal of the American Statistical Association*, 94(447):807–823, September 1999.
111. Santashil Pal Chaudhuri, Rajnish Kumar, and Amit Kumar Saha. A MAC protocol for multi-frequency physical layer. Technical report, Rice University, Houston, TX, USA, January 2003.
112. Kameswari Chebrolu, Bhaskaran Raman, and Sayandeep Sen. Long-distance 802.11b links: performance measurements and experience. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 74–85, Los Angeles, California, USA, September 2006.

113. Minghua Chen and Avidesh Zakhor. Flow control over wireless network and application layer implementation. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
114. Chun cheng Chen, Eunsoo Seo, Hwangnam Kim, and Haiyun Luo. Self-learning collision avoidance for wireless networks. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
115. Francisco Chinchilla, Mark Lindsey, and Maria Papadopouli. Analysis of wireless information locality and association patterns in a campus. In *IEEE Conference on Computer Communications (InfoCom)*, Hong Kong, March 2004.
116. Krishna Chintalapudi, Ramesh Govindan, Gaurav Sukhatme, and Amit Dhariwal. Ad-hoc localization using ranging and sectoring. In *IEEE Conference on Computer Communications (InfoCom)*, Hong Kong, March 2004.
117. Sunwoong Choi, Kihong Park, and Chong kwon Kim. On the performance characteristics of WLANs: revisited. In *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 97–108, Banff, Alberta, Canada, June 2005.
118. Chun-Ting Chou, S. N. Shankar, and Kang G. Shin. Achieving per-stream QoS with distributed airtime allocation and admission control in IEEE 802.11e wireless LANs. In *IEEE Conference on Computer Communications (InfoCom)*, pages 1584–1595, Miami, Florida, USA, March 2005.
119. Bent Guldberg Christensen. Lightpeers: A lightweight mobile p2p platform. In *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '07)*, pages 132–136, White Plains, NY, March 2007.
120. William S. Cleveland, Dong Lin, and Don X. Sun. IP packet generation: statistical models for TCP start times based on connection-rate superposition. In *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 166–177, Santa Clara, CA, United States, June 2000.
121. IEEE Computer Society LANMAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Standard 802.11-1999, New York, NY, USA, 1999.
122. Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, CA, September 2003.
123. Mark Crovella and Azer Bestavros. Self-similarity in world-wide-web traffic: Evidence and possible causes. In *Proceedings of SIGMETRICS*, Philadelphia, PA, May 1996.
124. Mark E. Crovella and Azer Bestavros. Self-similarity in world-wide-web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, December 1997.
125. Ralph B. D'Agostino and Michael A. Stephens. *Goodness-of-Fit Techniques*. Marcel Dekker, 1986.
126. Samir Das, Charles Perkins, and Elizabeth Royer. Performance comparison of two on-demand routing protocols for ad-hoc networks. In *IEEE Conference on Computer Communications (InfoCom)*, Tel Aviv, Israel, March 2000.
127. James Davis, Andy Fagg, and Brian Neil Levine. Wearable computers as packet transport mechanisms in highly partitioned ad-hoc networks. In *Proc. International Symposium on Wearable Computers (ISWC)*, Zurich, October 2001.

128. Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, 1992.
129. R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, USA, September 2004.
130. Richard Draves, Jitendra Padhye, and Brian Zill. Comparison of routing metrics for static multi-hop wireless networks. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, Portland, OR, USA, August 2004.
131. Richard Durrett. *Lecture notes on particle systems and percolation*. Pacific Grove, CA, 1988.
132. Bradley M. Duska, David Marwood, and Michael J. Feeley. The measured access characteristics of world-wide-web client proxy caches. In *USENIX Symposium on Internet Technologies and Systems, 1997*, Monterey, CA, December 1997.
133. Nathan Eagle and Alex (Sandy) Pentland. CRAWDAD data set mit/reality (v. 2005-07-01). Downloaded from <http://crawdad.cs.dartmouth.edu/mit/reality>, July 2005.
134. David Eckhardt and Peter Steenkiste. Measurement and analysis of the error characteristics of an in building wireless network. *ACM Computer Communication Review*, 26(4):243–254, October 1996.
135. Shane Eisenman. CRAWDAD data set columbia/ecsma (v. 2006-11-17). Downloaded from <http://crawdad.cs.dartmouth.edu/columbia/ecsma>, November 2006.
136. Shane Eisenman and Andrew Campbell. E-CSMA: Supporting enhanced CSMA performance in experimental sensor networks using per-neighbor transmission probability thresholds. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, Anchorage, AL, May 2007.
137. Chip Elliott. Building the Wireless Internet. *IEEE Spectrum*, 38(1), January 2001.
138. Mike Esler, Jeffrey Hightower, Tom Anderson, and Gaetano Borriello. Next century challenges: data-centric networking for invisible computing – the Portolano Project at the University of Washington. In *Proceedings of the Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 256–262, Seattle, Washington, August 1999.
139. Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In ACM, editor, *Proceedings of the Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 263–270, Seattle, Washington, August 1999.
140. Patrick T. Eugster, Rachid Guerraoui, Anne-Marie Kermarrec, and Laurent Massoulié. Epidemic information dissemination in distributed systems. *IEEE Computer*, 37(5):60–67, 2004.
141. Frederic Evennou and Francois Marx. Improving positioning capabilities for indoor environments with WiFi. In *EUSIPCO 2005*, Antalya, Turkey, September 2005. IST.
142. Frederic Evennou and Francois Marx. Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. In *EURASIP Journal on Applied Signal Processing*, pages pp. 1–11, 2006.

143. Johannes F. Network game traffic modelling. In *NetGames '02: Proceedings of the 1st workshop on Network and system support for games*, pages 53–57, New York, NY, USA, 2002.
144. Kevin Fall and Kannan Varadhan. ns: Notes and Documentation. Technical report, University of California at Berkeley, LBL, USC/ISI, and Xerox PARC, October 1998.
145. Jianqing Fan and Irene Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman and Hall, London, 1996.
146. Lei Fang, Wenliang Du, and Peng Ning. A beacon-less location discovery scheme for wireless sensor networks. In *IEEE Conference on Computer Communications (InfoCom)*, pages 161–171, Miami, Florida, March 2005.
147. Laura Marie Feeney. Investigating the energy consumption of an IEEE 802.11 network interface. Technical Report SICS-T 99/11-SE, Swedish Institute of Computer Science, December 1999.
148. Silke Feldmann, Kyandoghere Kyamakya, Ana Zapater, and Zighuo Lue. An indoor Bluetooth-based positioning system: concept, implementation and experimental evaluation. In *International Conference on Wireless Networks*, pages 109–113, Las Vegas, Nevada, USA, 2003.
149. S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, August 2001.
150. Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, August 1993.
151. Charalampos Fretzagias and Maria Papadopouli. Cooperative Location Sensing for Wireless Networks. In *Second IEEE International conference on Pervasive Computing and Communications*, Orlando, Florida, March 2004.
152. Traces from CDMA 1x EV-DO Network.
<http://networks.cnu.ac.kr/measurement/cdma-1x-evdo/>.
153. Richard M. Fujimoto, Randall Guensler, Michael P. Hunter, Hao Wu, Mahesh Palekar, Jaesup Lee, and Joonho Ko. CRAWDAD data set gatech/vehicular (v. 2006-03-15). Downloaded from <http://crawdad.cs.dartmouth.edu/gatech/vehicular>, March 2006.
154. Yan Gao, Jennifer Hou, and Hoang Nguyen. Topology Control for Maintaining Network Connectivity and Maximizing Network Capacity under the Physical Model. In *IEEE Conference on Computer Communications (InfoCom)*, Phoenix, Arizona, USA, April 2008.
155. Michele Garetto, Jingpu Shi, and Edward W. Knightly. Modeling media access in embedded two-flow topologies of multi-hop wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 200–214, Cologne, Germany, August 2005.
156. Richard Gass, James Scott, and Christophe Diot. CRAWDAD data set cambridge/inmotion (v. 2005-10-01). Downloaded from <http://crawdad.cs.dartmouth.edu/cambridge/inmotion>, October 2005.
157. Mario Gerla and Jack Tsai. Multicluster, mobile, multimedia radio network. *Journal of Wireless Networks*, 1(3):255–265, 1995.
158. Christos Gkantsidis and Pablo Rodriguez. Network coding for large-scale content distribution. In *IEEE Conference on Computer Communications (InfoCom)*, Miami, FL, USA, March 2005.
159. Gnutella. <http://gnutella.wego.com>.

160. Sanjay Goel and Stephen Bush. Biological models of security for virus propagation in computer networks. *Login*, 29(6), December 2004.
161. Tom Goff, Nael B. Abu-Ghazaleh, Dhananjay S. Phatak, and Ridvan Kahvecioglu. Preemptive routing in ad hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 43–52, Rome, Italy, July 2001.
162. Nina Golyandina, Vladimir Nekrutkin, and Anatoly Zhigljavsky. *Analysis of Time Series Structure: SSA and Related Techniques*. Chapman & Hall/CRC, 2001.
163. Amal Graafstra. Hands on: How radio-frequency identification and I got personal. *IEEE Spectrum*, March 2007.
164. Boris Grondahl. Wireless world meets to lick its wounds.
<http://www.thestandard.com/article/display/0,1151,22322,00.html?nl=dnt>.
165. Bjorn Gronvall, Assar Westerlund, and Stephen Pink. The design of a multicast-based distributed file system. In *Third Symposium on Operating Systems Design and Implementation*, New Orleans, LA, USA, February 1999.
166. Matthias Grossglauser and Patrick Thiran. Networks out of control: models and methods for random networks. Technical report, School of Computer and Communication Sciences, Ecole Polytechnique Federale de Lausanne (EPFL), 2005.
167. Matthias Grossglauser and David Tse. Mobility increases the capacity of mobile ad-hoc wireless networks. In *IEEE Conference on Computer Communications (InfoCom)*, Anchorage, Alaska, April 2001.
168. Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of peer-to-peer file sharing workload. In *ACM Symposium on Operating Systems Principles*, October 2003.
169. Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang. Measurements, analysis, and modeling of BitTorrent-like systems. In *ACM Internet Measurement Conference (Sigcomm)*, Philadelphia, PA, USA, August 2005.
170. Piyush Gupta and P.R. Kumar. The capacity of wireless networks. *Transactions of Information Theory*, 46(2):388–404, March 2000.
171. Youngjune Gwon, Ravi Jain, and Toshiro Kawahara. Robust indoor location estimation of stationary and mobile users. In *IEEE Conference on Computer Communications (InfoCom)*, March 2004.
172. Iyaylo Haratcherev, Koen Langendoen, Reginald Lagendijk, and Henk Sips. Hybrid rate control for IEEE 802.11. In *Proceedings of the second international workshop on Mobility management & wireless access protocols (MobiWac)*, pages 10–18, New York, NY, USA, September 2004.
173. Iyaylo Haratcherev, Jacco Taal, Koen Langendoen, Reginald Lagendijk, and Henk Sips. Fast 802.11 link adaptation for real-time video streaming by cross-layer signalling. In *IEEE Symposium in Circuits and Systems (ISCAS)*, Kobe, May 2005.
174. Shlomo Havlin, Menachem Dishon, James E. Kiefer, and George H. Weiss. Trapping of random walk in two and three dimensions. *Physical Review Letter*, 53(5):407–410, July 1984.
175. Simon Haykin. Cognitive radio: Brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, February 2005.

176. Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large-scale sensor networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, CA, USA, September 2003.
177. Tristan Henderson, David Kotz, and Ilya Abyzov. The changing usage of a mature campuswide wireless network. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, USA, September 2004.
178. Félix Hernández-Campos. *Generation and Validation of Empirically-Derived TCP Application Workloads*. PhD thesis, University of North Carolina at Chapel Hill, 2006.
179. Félix Hernández-Campos, Merkourios Karaliopoulos, Maria Papadopouli, and Haipeng Shen. Spatio-temporal modeling of traffic workload in a campus WLAN. In *Second annual international Wireless Internet Conference*, Boston, USA, August 2006.
180. Félix Hernández-Campos and Maria Papadopouli. Assessing The Real Impact of 802.11 WLANs: A Large-Scale Comparison of Wired and Wireless Traffic. In *14th IEEE Workshop on Local and Metropolitan Area Networks*, Chania, Greece, September 2005.
181. Félix Hernández-Campos and Maria Papadopouli. A comparative measurement study of the workload of wireless access points in campus networks. In *16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Berlin, Germany, September 2005.
182. Félix Hernández-Campos and Maria Papadopouli. A comparative measurement study of the workload of wireless access points in campus networks. Technical Report 353, ICS-FORTH, Heraklion, Greece, March 2005.
183. Jeffrey Hightower and Gaetano Borriello. A Survey and Taxonomy of Location Sensing Systems for Ubiquitous Computing. Technical Report, University of Washington, Department of Computer Science and Engineering UW CSE 01-08-03, Seattle, WA, August 2001.
184. Jeffrey Hightower and Gaetano Borriello. Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study. In *Proceedings of the Sixth International Conference on Ubiquitous Computing (UbiComp)*, Nottingham, England, September 2004.
185. Jeffrey Hightower, Roy Want, and Gaetano Borriello. SpotON: An indoor 3D location sensing technology based on RF signal strength. UW CSE tech report 2000-02-02, University of Washington, Seattle, WA, February 2000.
186. Gavin Holland, Nitin Vaidya, and Paramvir Bahl. A rate-adaptive MAC protocol for multi-hop wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 236–251, Rome, Italy, July 2001.
187. T. Horozov, A. Grama, V. Vasudevan, and S. Landis. Moby — a mobile peer-to-peer service and data network. In *Proceedings of International Conference on Parallel Processing*, pages 437–444, Washington, DC, USA, August 2002.
188. A. Howard, S. Siddiqi, and G. Sukhatme. An experimental study of localization using wireless ethernet. In *International Conference on Field and Service Robotics*, Yamanaka, Japan, July 2003.
189. Wei-Jen Hsu, Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Ahmed Helmy. Modeling time-variant user mobility in wireless mobile networks. Anchorage, Alaska, USA, May 2007.

190. Chunyu Hu and Jennifer C. Hou. A novel approach to contention control in IEEE 802.11e-operated WLANs. In *IEEE Conference on Computer Communications (InfoCom)*, pages 1190–1198, Anchorage, Alaska, USA, May 2007.
191. Lingxuan Hu and David Evans. Localization for mobile sensor networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2004.
192. Yih-Chun Hu and David B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 231–242, Boston, MA, USA, August 2000.
193. Yih-Chun Hu and David B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 231–242, Boston, MA, USA, August 2000.
194. Yih-Chun Hu and David B. Johnson. Implicit source routes for on-demand ad-hoc network routing. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 1–10, New York, NY, USA, October 2001.
195. Jean-Pierre Hubaux, Levente Buttan, and Srdan Capkun. The quest for security in mobile ad-hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 146–155, Long Beach, CA, October 2001.
196. Barry D. Hughes. *Random Walks and Random Environments*. Oxford Science Publications, 1995.
197. Wing-Chung Hung, K.L. Eddie Law, and A. Leon-Garcia. A dynamic multi-channel MAC for ad-hoc LAN. In *Proceedings of the 21st Biennial Symposium on Communications*, Kingston, Canada, June 2002.
198. Tomasz Imielinski, S. Viswanathan, and B. R. Badrinath. Energy efficient indexing on air. In *ACM SIGMOD International conference on Management of Data*, Minneapolis, MN, USA, May 1994.
199. Mikel Izal, Guillaume Urvoy-Keller, Ernst Biersack, Pascal Felber, Anwar Hamra, and Luis Garcés-Erice. Dissecting BitTorrent: Five months in a torrent’s lifetime. In *5th Passive and Active Measurement Workshop*, Antibes Juan-les-Pines, France, April 2004.
200. Nitin Jain and Samir Das. A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In *Proceedings of the 9th Int. Conf. On Computer Communications and Networks (IC3N)*, Phoenix, USA, October 2001.
201. Ravi Jain, Dan Lelescu, and Mahadevan Balakrishnan. Model T: an empirical model for user registration patterns in a campus wireless LAN. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Cologne, Germany, August 2005.
202. Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a delay-tolerant network. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, Portland, OR, USA, August 2004.
203. Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad-hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, San Diego, CA, September 2003.

204. Amit Jardosh, Krishna Ramachandran, Kevin Almeroth, and Elizabeth MBelding-Royer. Understanding congestion in IEEE 802.11b wireless networks. In *Proceedings of the Internet Measurement Conference*, Berkeley, CA, USA, October 2005.
205. Amit Jardosh, Krishna N. Ramachandran, Kevin C. Almeroth, and Elizabeth Belding. CRAWDAD data set ucsb/ietf2005 (v. 2005-10-19). Downloaded from <http://crawdad.cs.dartmouth.edu/ucsb/ietf2005>, October 2005.
206. Amit P. Jardosh, Kimaya Mittal, Krishna N. Ramachandran, Elizabeth M. Belding, and Kevin C. Almeroth. IQU: practical queue-based user association management for WLANs. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 158–169, Los Angeles, California, USA, September 2006.
207. Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek, and Mikael Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 195–206, Seattle, Washington, USA, August 1999.
208. Ian T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
209. Julien Jomier. Note sharing application, 2003.
210. Evan P. C. Jones, Lily Li, and Paul A. S. Ward. Practical routing in delay-tolerant networks. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, pages 237–243, Philadelphia, Pennsylvania, USA, August 2005.
211. Youngmi Joo, Vinay Ribeiro, Anja Feldmann, Anna C. Gilbert, and Walter Willinger. Tcp/ip traffic dynamics and network performance: a lesson in workload modeling, flow control, and trace-driven simulations. *ACM Computer Communication Review*, (2), 2001.
212. Julie Letchner and Dieter Fox and Anthony LaMarca. Large-Scale Localization from Wireless Signal Strength. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, Pittsburgh, Pennsylvania, USA, July 2005.
213. M. Kamenetsky and M. Unbehaun. Coverage Planning for Outdoor Wireless LAN Systems. In *IEEE International Zurich Seminar on Broadband Communications*, Zurich, Switzerland, February 2002.
214. Ad Kamerman and Leo Monteban. Wavelan(c)-ii: a high-performance wireless lan for the unlicensed band. *Bell Labs Technical Journal*, 2(3):118–133, 1997.
215. Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc Claffy. Transport layer identification of p2p traffic. In *ACM Sigcomm Internet Measurement Conference*, San Diego, CA, USA, October 2004.
216. Thomas Karagiannis, Konstantina Papagiannaki, and Michalis Faloutsos. BLINC: Multi-level Traffic Classification in the Dark. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, Philadelphia, PA, USA, August 2005.
217. Merkouris Karaliopoulos, Maria Papadopoulou, Elias Raftopoulos, and Haipeng Shen. On scalable measurement-driven modelling of traffic demand in large WLANs. In *IEEE Workshop on Local and Metropolitan Area Networks*, Princeton NJ, USA, June 2007.

218. Brad Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 243–254, Boston, MA, USA, 2000.
219. Anand Kashyap, Samrat Ganguly, and Samir R. Das. A measurement-based approach to modeling link capacity in 802.11-based wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 242–253, Montreal, Quebec, Canada, September 2007.
220. Anastasia Katranidou and Maria Papadopoulou. Location-sensing Using the IEEE 802.11 Infrastructure and the Peer-to-Peer Paradigm for Mobile Computing Applications. Master’s thesis, University of Crete, Heraklion, Greece, February 2006.
221. Harry Kesten and Vidas Sidoravicius. A shape theorem for the spread of an infection. *Preprint: math.PR/0312511 at arXiv.org. Math.*
222. Jongseok Kim, Seongkwan Kim, Sunghyun Choi, and Daji Qiao. CARA: Collision-aware rate adaptation for IEEE802.11 WLANs. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
223. Minkyong Kim and David Kotz. Modeling users’ mobility among WiFi access points. In *Proceedings of the International Workshop on Wireless Traffic Measurements and Modeling*, Seattle, WA, June 2005. USENIX Association.
224. Minkyong Kim and David Kotz. Periodic properties of user mobility and AP popularity. *Journal of Personal and Ubiquitous Computing*, 11(6), August 2007. Special Issue of papers from LoCA 2005.
225. Tae-Suk Kim, Hyuk Lim, and Jennifer Hou. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Los Angeles, California, USA, September 2006.
226. James J. Kistler and M. Satyanarayanan. Disconnected operation in the Coda file system. *Proc. ACM Symposium on Operating Systems Principles*, 8(2):213–225, October 1991.
227. Jon Kleinberg. The wireless epidemic. *Nature (News and Views)*, 449:287–288, 2007.
228. Young-Bae Ko and Nitin H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, October 1998.
229. Can Emre Koksal, Kyle Jamieson, Emre Telatar, and Patrick Thiran. Impacts of channel variability on link-level throughput in wireless networks. In *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, pages 51–62, Saint Malo, France, June 2006.
230. Gerd Kortuem. Proem: a middleware platform for mobile peer-to-peer computing. *SIGMOBILE Mobile Computing and Communications Review*, 6(4):62–64, October 2002.
231. Niko Kotilainen and Maria Papadopoulou. You’ve got photos! the design and evaluation of a location-based media-sharing application. In *4th International Mobile Multimedia Communications Conference (Mobimedia)*, Oulu, Finland, July 2008.
232. Niko Kotilainen, Matthieu Weber, Mikko Vapa, and Jarkko Vuori. Mobile Chedar - a peer-to-peer middleware for mobile devices. In *Proceedings of the Second International Workshop on Mobile Peer-to-Peer Computing (MP2P’05)*, pages 86–90, Kauai Island, Hawaii, March 2005.

233. David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. Technical Report TR2002-432, Dept. of Computer Science, Dartmouth College, September 2002.
234. David Kotz and Kobby Essien. Analysis of a campus-wide wireless network. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 107–118, September 2002. Revised and corrected as Dartmouth CS Technical Report TR2002-432.
235. Iordanis Koutsopoulos and Leandros Tassiulas. Joint optimal access point selection and channel assignment in wireless networks. *IEEE/ACM Transactions on Networking*, (3), June 2007.
236. John Krumm, Steve Harris, Brian Meyers, Barry Brumitt, Michael Hale, and Steve Shafer. Multi-Camera Multi-Person Tracking for EasyLiving. In *Proceedings of the Third IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, July 2000.
237. Anurag Kumar, Eitan Altman, Daniele Miorandi, and Munish Goyal. New insights from a fixed point analysis of single cell IEEE802.11 WLANs. In *IEEE Conference on Computer Communications (InfoCom)*, pages 1550–1561, Miami, Florida, USA, March 2005.
238. Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. IEEE 802.11 rate adaptation: a practical approach. In *The Seventh ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM)*, pages 126–134, Venice, Italy, October 2004.
239. A. M. Ladd, K.E. Bekris, A. Rudys, G. Marceau, L. E. Kavraki, and D.S. Wallach. Robotics-Based Location Sensing using Wireless Ethernet. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Atlanta, GE, USA, September 2002.
240. Youngseok Lee, Kyoungae Kim, and Yanghee Choi. Optimization of AP placement and channel assignment in wireless LANs. In *IEEE Conference on Local Computer Networks*, Florida, FL, USA, November 2002.
241. Jeremie Leguay, Anders Lindgren, James Scott, Timur Friedman, Jon Crowcroft, and Pan Hui. CRAWDAD data set upmc/content (v. 2006-11-17). Downloaded from <http://crawdad.cs.dartmouth.edu/upmc/content>, November 2006.
242. Hui Lei and Dan Duchamp. An analytical approach to file prefetching. In *USENIX Annual Technical Conference*, Anaheim, CA, January 1997.
243. Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM Symposium on Communications Architectures and Protocols (SigComm)*, pages 183–193, San Francisco, California, September 1993. ACM. also in *Computer Communication Review* 23 (4), Oct. 1992.
244. H. Leung, T. Lo, and S. Wang. Prediction of noisy chaotic time series using an optimal radial basis function neural network. *IEEE Transactions on Neural Networks*, 12(5):1163–1172, September 2001.
245. Peter A.W. Lewis and Gerald S. Shedler. Simulation of nonhomogeneous Poisson process by thinning. *Naval Research Logistics Quarterly*, 26:403–413, 1979.
246. Jiandong Li, Zygmunt J. Haas, Min Sheng, and Yanhui Chen. Performance evaluation of modified IEEE 802.11 MAC for multi-channel multi-hop ad-hoc network. In *17th International Conference on Advanced Information Networking and Applications (AINA)*, Xi'an, China, March 2003.

247. Li Li, Joseph Halpern, Paramvir Bahl, Yi-Min Wang, and Roger Wattenhofer. Analysis of a cone-based distributed topology control algorithm for wireless multi-hop networks. In *PODC '01*, Newport, Rhode Island, August 2001.
248. Lun Li, David Alderson, John Doyle, and Walter Willinger. Towards a theory of scale-free graphs: definition, properties, and implications. *Internet Mathematics*.
249. Ning Li and Jennifer Hou. FLSS: A Fault-Tolerant Topology Control. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Philadelphia, Pennsylvania, September 2004.
250. Ning Li and Jennifer Hou. Localized topology control algorithms for heterogeneous wireless networks. *IEEE/ACM Transactions On Networking*, 13(6), December 2005.
251. Ning Li, Jennifer Hou, and Lui Sha. Design and analysis of an mst-based topology control algorithm. *IEEE Transactions on Wireless Communications*, 4(3), May 2005.
252. Qun Li and Daniela Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 44–55, Boston, MA, USA, August 2000.
253. T. Lindeberg. *Scale Space Theory in Computer Vision*. Kluwer, Boston, 1994.
254. Christoph Lindemann and Oliver P. Waldhorst. Modeling epidemic information dissemination on mobile devices with finite buffers. In *ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems*, Banff, Alberta, Canada, June 2005.
255. Mark Lindsey. Correlations among nearby mobile web users. Master's thesis, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, May 2003.
256. The Jakarta Project Lucene. <http://jakarta.apache.org/lucene/docs/index.html>.
257. Henrik Lundgren, Krishna Ramachandran, Elizabeth Belding-Royer, Kevin Almeroth, Michael Benny, Andrew Hewatt, Alexander Touma, and Amit Jardosh. Experiences from the design, deployment, and usage of the UCSB Mesh-Net testbed. *IEEE Wireless Communications*, 44(4):18–29, April 2006.
258. J. S. Marron, Félix Hernández-Campos, and F. D. Smith. A sizer analysis of IP flow start times. *Institute of Mathematical Statistics Lecture Notes-Monograph Series, J. Rojo and V. Perez-Abreu (Eds)*, 44:87–105, 2004.
259. Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad-hoc networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 255–265, Boston, MA, USA, August 2000.
260. Marvin McNett and Geoffrey M. Voelker. Access and mobility of wireless PDA users. *Mobile Computing Communications Review*, 9(2):40–55, April 2005.
261. Xiaojiao Meng, Starsky Wong, Yuan Yuan, and Songwu Lu. Characterizing flows in large wireless data networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 174–186, Philadelphia, PA, September 2004.
262. Dejan Milojevic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-peer computing. Technical Report HPL-2002-57, HP Laboratories, Palo Alto, USA, March 2002.
263. Arunesh Mishra, Vladimir Brik, Suman Banerjee, Aravind Srinivasan, and William Arbaugh. A client-driven approach for channel management in wire-

- less LANs. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
264. Arunesh Mishra, Minh Shin, and William A. Arbaugh. An empirical analysis of the IEEE 802.11 MAC layer handoff process. *ACM SIGCOMM Computer Communication Review*, pages 93–102, April 2003.
 265. Joseph Mitola. *Cognitive radio: An integrated agent architecture for software defined radio*. PhD thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, May 2000.
 266. Michael Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet Mathematics*, 2003.
 267. Allen K. L. Miu, Hari Balakrishnan, and Can Emre Koksul. Improving loss resilience with multi-radio diversity in wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 16–30, Cologne, Germany, August 2005.
 268. A. Moore and K. Papagiannaki. Toward the Accurate Identification of Network Applications. In *Passive and Active Measurement Workshop*, Boston, MA, USA, March 2005.
 269. Thomas Moscibroda, Roger Wattenhofer, and Aaron Zollinger. Topology Control Meets SINR: The Scheduling Complexity of Arbitrary Topologies. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Florence, Italy, May 2006.
 270. Mirco Musolesi and Cecilia Mascolo. A community based mobility model for ad hoc network research. In *Second International Workshop on Multi-hop Ad Hoc Networks (REALMAN)*, May 2006.
 271. Tamer Nadeem, Lusheng Ji, Ashok K. Agrawala, and Jonathan R. Agre. Location enhancement to IEEE802.11 DCF. In *IEEE Conference on Computer Communications (InfoCom)*, pages 651–663, Miami, Florida, USA, March 2005.
 272. Napster. <http://www.napster.com>.
 273. Asis Nasipuri and Samir R. Das. A multichannel CSMA MAC protocol for mobile multihop networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, LA, USA, September 1999.
 274. Huu Quynh Nguyen, François Baccelli, and Daniel Kofman. A stochastic geometry analysis of dense IEEE802.11 networks. In *IEEE Conference on Computer Communications (InfoCom)*, pages 1199–1207, Anchorage, Alaska, USA, May 2007.
 275. Drago Niculescu and Badri Nath. Ad Hoc Positioning System (APS). In *IEEE Conference on Global Communications (GLOBECOM)*, San Antonio, TX, November 2001.
 276. Dragos Niculescu and Badri Nath. Ad Hoc Positioning System (APS) using AoA. In *IEEE Conference on Computer Communications (InfoCom)*, San Francisco, CA, April 2003.
 277. Brian D. Noble and Mahadev Satyanarayanan. Experience with adaptive mobile applications in odyssey. *Mobile Networks and Applications*, 4(4):245–254, December 1999.
 278. Carl Nuzman, Iraj Saniee, Wim Sweldens, and Alan Weiss. A compound model for TCP connection arrivals for LAN and WAN applications. *Computer Networks*, 40(3):319–337, October 2002.

279. The New York Times on the Web.
http://www.nytimes.com/adinfo/wireless_audience.html.
280. A. Ovchinnikov, S. Timashev, and A. Belyy. *Kinetics of Diffusion controlled chemical processes*. Nova Science Publishers, 1989.
281. Thomas W. Page, Richard G. Guy, John S. Heidemann, David Ratner, Peter Reiher, Ashish Goel, Geoffrey H. Kuenning, and Gerald J. Popek. Perspectives on optimistically replicated peer-to-peer filing. *Software—Practice and Experience*, 28(2):155–180, February 1998.
282. Maria Papadopoulou, Elias Raftopoulos, and Haipeng Shen. Evaluation of short-term traffic forecasting algorithms in wireless networks. In *2nd Conference on Next Generation Internet Design and Engineering*, Valencia, Spain, April 2006.
283. Maria Papadopoulou and Henning Schulzrinne. Seven degrees of separation in mobile ad hoc networks. In *IEEE Conference on Global Communications (GLOBECOM)*, San Francisco, CA, November 2000.
284. Maria Papadopoulou and Henning Schulzrinne. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Long Beach, CA, October 2001.
285. Maria Papadopoulou and Henning Schulzrinne. A performance analysis of 7DS a peer-to-peer data dissemination and prefetching tool for mobile users. In *Advances in wired and wireless communications, IEEE Sarnoff Symposium Digest*, Ewing, NJ, March 2001.
286. Maria Papadopoulou and Henning Schulzrinne. Performance of data dissemination and message relaying in mobile ad hoc networks. Technical Report CUCS-004-02, Dept. of Computer Science, Columbia University, New York, NY, February 2002.
287. Maria Papadopoulou, Haipeng Shen, Elias Raftopoulos, Manolis Ploumidis, and Félix Hernández-Campos. Short-term traffic forecasting in a campus-wide wireless network. In *16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications*, Berlin, Germany, September 2005.
288. Maria Papadopoulou, Haipeng Shen, and Manolis Spanakis. Characterizing the duration and association patterns of wireless access in a campus. In *11th European Wireless Conference*, Nicosia, Cyprus, April 2005.
289. Maria Papadopoulou, Haipeng Shen, and Manolis Spanakis. Modeling client arrivals at access points in wireless campus-wide networks. In *14th IEEE Workshop on Local and Metropolitan Area Networks*, Chania, Greece, September 2005.
290. Maria Papadopoulou, Haipeng Shen, and Manolis Spanakis. Modeling client arrivals at access points in wireless campus-wide networks. Technical Report 357, FORTH-ICS, Heraklion, Greece, May 2005.
291. Konstantina Papagiannaki, Mark Yarvis, and W. Steven Conner. CRAWDAD data set intel/home (v. 2006-04-16). Downloaded from <http://crawdad.cs.dartmouth.edu/intel/home>, April 2006.
292. Konstantina Papagiannaki, Mark D. Yarvis, and W. Steven Conner. Experimental characterization of home wireless networks and design implications. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
293. Theodore Patkos, Antonis Mpikakis, Grigoris Antoniou, Maria Papadopoulou, and Dimitris Plexousakis. A semantic-based framework for context-aware

- pedestrian guiding services. In *Second International Workshop on Semantic Web Technology For Ubiquitous and Mobile Applications*, Trentino, Italy, 2006.
294. Vern Paxson. Empirically-derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, August 1994.
 295. Vern Paxson and Sally Floyd. Wide-area traffic: the failure of Poisson modeling. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, pages 257–268, London, United Kingdom, August 1994.
 296. Vern Paxson and Sally Floyd. Wide-area traffic: The failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3), June 1995.
 297. Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM Conference on Communications Architectures, Protocols and Applications*, volume 24, pages 234–244, October 1994.
 298. Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, and Mahesh K. Marina. Performance comparison of two on-demand routing protocols for ad-hoc networks. *IEEE Personal Communications Magazine*, 8(1), February 2001.
 299. Roman Pichna, Tero Ojanpera, Harri Posti, and Jouni Karppinen. Wireless Internet - IMT-2000/wireless LAN Interworking. *Journal of Communications and Networking*, 2(1):46–57, March 2000.
 300. Manolis Ploumidis, Maria Papadapouli, and Thomas Karagiannis. Multi-level application-based traffic characterization in a large-scale wireless network. In *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Helsinki, Finland, June 2007.
 301. Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 32–43, Boston, MA, USA, August 2000.
 302. Nissanka B. Priyantha, Allen K. L. Miu, Hari Balakrishnan, and Seth Teller. The Cricket compass for context-aware mobile applications. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–14, Rome, Italy, July 2001.
 303. DATAMAN project. <http://www.cs.rutgers.edu/dataman/>, 2001.
 304. D. Qiao and S. Choi. Fast-responsive link adaptation for IEEE802.11 WLANs. In *IEEE International Conference on Communications (ICC)*, Seoul, Korea, May 2005.
 305. Daji Qiao, Sunghyun Choi, and Kang G. Shin. Goodput analysis and link adaptation for IEEE 802.11a wireless LANs. *IEEE Transactions on Mobile Computing*, 1(4):278–292, 2002.
 306. Chandra R., Lili Qiu, Jain K., , and M. Mahdian. Optimizing the placement of Internet taps in wireless neighborhood networks. In *International Conference on Network Protocols (ICNP)*, Berlin, Germany, October 2004.
 307. R. Harle, A. Ward, and A. Hopper. Single reflection spatial voting. In *Proceedings of the First International Conference on Mobile Systems, Applications, and Services*, San Francisco, May 2003.
 308. Krishna Ramachandran, Elizabeth Belding, Kevin Almeroth, and Milind Budhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
 309. Ram Ramanathan. A unified framework and algorithm for channel assignment in wireless networks. *Wireless Networks*, 5(2):81–94, March 1999.

310. Ishwar Ramani and Stefan Savage. SyncScan: Practical fast handoff for 802.11 infrastructure networks. In *IEEE Conference on Computer Communications (InfoCom)*, Miami, FL, USA, March 2005.
311. Theodore S. Rappaport. *Wireless Communications: Principles and Practice*. IEEE Press, New York, 1996.
312. Krishnamurthi Ravishankar and Suresh Singh. Broadcasting on $[0, L]$. *Discreet Applied Mathematics*, 53(1-3):299–320, 1994.
313. Krishnamurthi Ravishankar and Suresh Singh. Asymptotically optimal gossiping on $[0, l]$. *Discreet Applied Math*, 1995.
314. Krishnamurthi Ravishankar and Suresh Singh. Central limit theorem for time to broadcast on $[0, l]$. *Probability in the Applied and Informational Sciences*, 9:201–209, 1995.
315. RIM. <http://www.goamerica.net/coverage/cingular.html>.
316. John Risson and Tim Moors. Survey of research towards robust peer-to-peer networks: Search methods. Technical report UNSW-EE-P2P-1-1, University of New South Wales, Sydney, Australia, September 2004.
317. Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, John Zahorjan, and Ed Lazowska. CRAWDAD data set uw/sigcomm2004 (v. 2006-10-17). <http://crawdad.cs.dartmouth.edu/uw/sigcomm2004>, October 2006.
318. Miguel Rodriguez, Juan P. Pece, and Carlos J. Escudero. In-building location using bluetooth. In *International Workshop on Wireless Ad-hoc Networks*, Coruna, Spain, May 2005.
319. Sheldon Ross. *Introduction to probability models*. Academic Press, London, 1993.
320. Sheldon Ross. *Stochastic Processes*. Jon Wiley and Sons, New York, 1996.
321. Sheldon M. Ross. *Stochastic Processes*. John Wiley and Sons, New York, New York, 1983.
322. Abhishek Roy, Archan Misra, and Sajal K. Das. An information theoretic framework for optimal location tracking in multi-system 4G wireless networks. In *IEEE Conference on Computer Communications (InfoCom)*, Hong Kong, March 2004.
323. Roy Want and Andy Hopper and Veronica Falcao and Jon Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.
324. Elizabeth M. Royer and Charles E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In ACM, editor, *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 207–218, Seattle, Washington, August 1999.
325. Bahareh Sadeghi, Vikram Kanodia, Ashutosh Sabharwal, and Edward Knightly. Opportunistic media access for multirate ad-hoc networks. In *ACM Proceedings of the 8th annual international conference on Mobile computing and networking (MobiCom)*, pages 24–35, New York, NY, USA, 2002.
326. Prince Samar, Marc R. Pearlman, and Zygmunt J. Haas. Hybrid routing: the pursuit of an adaptable and scalable routing framework for ad-hoc networks. In *The handbook of ad hoc wireless networks*, pages 245–262. CRC Press, Inc., Boca Raton, FL, USA, December 2003.
327. Chris Savarese, Jan Rabaey, and Koen Langendoen. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *Proceedings of Usenix Annual Technical Conference*, Monterey, CA, June 2002.

328. Bruce Schneier. *Applied Cryptography*. John Wiley and Sons, 1995.
329. Vinay Seshadri, Gergely V. Zaruba, and Manfred Huber. A Bayesian sampling approach to indoor localization of wireless devices using received signal strength indication. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Kauai Island, Hawaii, March 2005.
330. Jungmin So and Nitin H. Vaidya. Multi-channel mac for ad-hoc networks: handling multi-channel hidden terminals using a single transceiver. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, pages 222–233, New York, NY, USA, 2004.
331. Joel Sommers, Hyungsuk Kim, and Paul Barford. Harpoon: a flow-level traffic generator for router and network tests. In *ACM SIGMETRICS poster session*, New York, NY, USA, 2004. ACM.
332. M. Spreitzer, M. Theimer, K. Petersen, A. Demers, and D. Terry. Dealing with server corruption in weakly consistent, replicated data systems. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 234–240, Budapest, Hungary, September 1997.
333. Sprint Applied Research Group.
<http://ipmon.sprintlabs.com/packstat/packetoverview.php>.
334. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, San Diego, CA, August 2001.
335. Jing Su and Stefan Saroiu. CRAWDAD data set toronto/bluetooth (v. 2006-08-29). Downloaded from <http://crawdad.cs.dartmouth.edu/toronto/bluetooth>, August 2006.
336. Karthikeyan Sundaresan and Konstantina Papagiannaki. The need for cross-layer information in access point selection algorithms. In *Internet Measurement Conference*, New York, NY, USA, 2006. ACM.
337. Carl Tait, Hui Lei, Swarup Acharya, and Henry Chang. Intelligent file hoarding for mobile computers. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Berkeley, CA, November 1995.
338. Diane Tang and Mary Baker. Analysis of a local-area wireless network. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 1–10, Boston, MA, USA, August 2000.
339. S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo Localization for Mobile Robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.
340. Cristian Tuduce and Thomas Gross. A mobility model based on wlan traces and its validation. In *IEEE Conference on Computer Communications (InfoCom)*, Miami, FL, USA, March 2005.
341. Kurt Tutschku. A measurement-based traffic profile of the edonkey filesharing service. In *Passive and Active Measurement Workshop*, Antibes Juan-les-Pins, France, April 2004.
342. George Tzagkarakis, Maria Papadapouli, and Panagiotis Tsakalides. Trend forecasting based on singular spectrum analysis of traffic workload in a large-scale wireless lan. January 2008.
343. George Tzagkarakis, Maria Papadapouli, and Panagiotis Tsakalides. Singular spectrum analysis of traffic workload in a large-scale wireless lan. In *10th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Chania, Crete, Greece, October 2007.

344. United Villages. <http://www.unitedvillages.com>.
345. Konstantinos Vandikas, Lito Kriara, Tonia Papakonstantinou, Anastasia Kastranidou, Haris Baltzakis, and Maria Papadopouli. Empirical-based analysis of a cooperative location-sensing system. In *ACM First International Conference on Autonomic Computing and Communication Systems (Autonomics)*, Rome, Italy, October 2007.
346. A. Vasan, R. Ramjee, and T. Woo. Echos-enhanced capacity 802.11 hotspots. In *IEEE Conference on Computer Communications (InfoCom)*, Miami, FL, USA, March 2005.
347. Arunchandar Vasan, Ramachandran Ramjee, and Thomas Y. C. Woo. ECHOS - enhanced capacity 802.11 hotspots. In *IEEE Conference on Computer Communications (InfoCom)*, pages 1562–1572, Miami, Florida, USA, March 2005.
348. S. Vasudevan, K. Papagiannaki, C. Diot, J. Kurose, and D. Towsley. Facilitating access point selection in ieee 802.11 wireless networks. In *Internet Measurement Conference*, Berkeley, CA, USA, 2005. USENIX Association.
349. Vindigo. http://www.vindigo.com/learn_more.html.
350. Kashi Venkatesh Vishwanath and Amin Vahdat. Realistic and responsive network traffic generation. In *ACM Symposium on Communications Architectures and Protocols (SigComm)*, Pisa, Italy, September 2006.
351. Roy Want and Andy Hopper. Active badges and personal interactive computing objects. Technical Report ORL 92-2, Olivetti Research, Cambridge, England, February 1992. also in *IEEE Transactions on Consumer Electronics*, Feb. 1992.
352. M. Weiser. The computer for the 21st century. *Scientific American*, September 1991.
353. Joshua Weitz, Philip Benfey, and Ned Wingreen. Evolution, interactions, and biological networks. *PLoS Biology*, 5(1), January 2007.
354. K. P. White. Simulating a nonstationary poisson process using bivariate thinning: The case of typical weekday arrivals at a consumer electronics store. *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future*, 1:458–461, 1999.
355. W. Willinger, Taqqu M.S., R. Sherman, and D.V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet LAN traffic at the source level. *ACM Computer Communication Review*, 25(4):100–113, October 1995.
356. W. Willinger, V. Paxson, and M. Taqqu. Self-similarity and heavy tails: Structural modeling of network traffic. In R. Adler, R. Feldman, and M. Taqqu, editors, *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, pages 27–53. Birkhauser, 1998.
357. Alec Wolman, Geoffrey M. Voelker, Nitin Sharma, Neal Cardwell, Anna Karlin, and Henry M. Levy. On the scale and performance of cooperative Web proxy caching. In *Proc. ACM Symposium on Operating Systems Principles*, Kiawah Island, SC, December 1999.
358. Starsky H. Y. Wong, Hao Yang, Songwu Lu, and Vaduvur Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 146–157, Los Angeles, California, USA, September 2006.
359. Starsky H.Y. Wong, Hao Yang, Songwu Lu, and Vaduvur Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM International Conference*

- on *Mobile Computing and Networking (MobiCom)*, Los Angeles, CA, September 2006.
360. Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad-hoc routing. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Rome, Italy, August 2001.
 361. J. Yang, C.-K. Lee Y. Chen, and M. Ammar. Ferry replacement protocols in sparse manet message ferrying systems. In *IEEE Wireless Communications and Networking (WCNC)*, New Orleans, LA, March 2005.
 362. Mark Yarvis, Konstantina Papagiannaki, and W. Steven Conner. Characterization of 802.11 wireless networks in the home. In *Proceedings of the First Workshop on Wireless Network Measurements (WiNMee)*, Trentino, Italy, April 2005.
 363. Tao Ye, H.-Arno Jacobsen, and Randy Katz. Mobile awareness in a wide area wireless network of info-stations. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, Texas, October 1998.
 364. J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *IEEE Conference on Computer Communications (InfoCom)*, San Francisco, CA, September 2003.
 365. Moustafa Youssef and Ashok Agrawala. The Horus WLAN location determination system. In *International conference on Mobile Systems, Applications and Services (MobiSys)*, Seattle, USA, June 2005.
 366. Moustafa Youssef, Adel Youssef, Chuck Rieger, Udaya Shankar, and Ashok Agrawala. PinPoint: An asynchronous time-based location determination system. In *International conference on Mobile Systems, Applications and Services (MobiSys)*, pages 165–176, Uppsala, Sweden, June 2006.
 367. Wenrui Zhao, Mostafa Ammar, and Ellen Zegura. A message ferrying approach for data delivery in sparse mobile ad-hoc networks. In *IEEE Conference on Computer Communications (InfoCom)*, Hong Kong, March 2004.
 368. Wenrui Zhao, Yang Chen, Mostafa Ammar, Mark D. Corner, Brian Neil Levine, and Ellen Zegura. Capacity Enhancement using Throwboxes in DTNs. In *IEEE International Conference on Mobile Ad hoc and Sensor Systems*, Vancouver, Canada, October 2006.
 369. Junlan Zhou, Zhengrong Ji, and Rajive Bagrodia. TWINE: A hybrid emulation testbed for wireless networks and applications. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.
 370. Lidong Zhou and Zygmunt J. Haas. Securing ad-hoc networks. *IEEE Network*, 13(6), November 1999.
 371. Jing Zhu, Benjamin Metzler, Xingang Guo, and York Liu. Adaptive CSMA for scalable network capacity in high-density WLAN: A hardware prototyping approach. In *IEEE Conference on Computer Communications (InfoCom)*, Barcelona, Spain, April 2006.

Index

- BLINC, 88
- FTP, 92
- HTTP traces, 81
- SNMP, 79, 81, 88
- SYSLOG, 79, 82
- SYSLOG messages, 82, 109
- TCP, 79
- 7DS, 13, 19
- 7DS host, 7DS node, 50
- 7DS node, 13
- 7DS prototype, 40
- 7DS simulation parameters, 53

- Active Badge, 39
- Active Bat, 31, 39
- active querying, 22, 51
- ad-hoc, 9
- advertisement, 50
- advertisements, 22
- AP, 5, 78
- AP path, 110
- AP-coresident client repeated request, 97
- AP-coresident-client repeated requests, 96
- application type, 88
- application-based classification, 88
- archives, 179
- arrival process of client visits, 125
- arrivals of clients, 123
- association process, 118
- asynchronous mode, 14, 24, 50, 53

- BiPareto distribution, 112

- BitTorrent, 8, 9
- building path, 110
- building types, 122, 142

- cache at each AP, 99
- cache attached to an AP, 96
- cache hierarchy, 99
- caching paradigms, 96
- campus-wide repeated requests, 96
- CBR, 106
- CCDF, 112
- channel selection, 169
- channel switching, 170
- chemotaxis, 173
- classify flows, 88
- client, 78
- client state, 83
- closed-loop traffic generation, 157
- CLS, 14, 19
- cognitive radio, 167
- collaborative location-sensing system, 14
- complementary cumulative distribution, 112
- connectivity problems, 166
- cooperation, 26
- CRAWDAD, 179
- Cricket, 31

- DAG, 80
- data dissemination, 68
- data repositories, 179
- data sharing (DS), 51

- data sharing and forwarding enabled (DS+FW), 51
- default querying mechanism, 22
- Default values in 7DS simulations, 54
- delay-tolerant, 11
- delay-tolerant networks, 15
- device mobility, 106, 172
- diffusion, 69, 173
- disconnections to the Internet, 12
- distance-prediction based positioning, 31
- dominant application, 92
- dominated, 87
- downloaders, 87
- drop-in client, 84
- DTN, 15
- dynamic port, 88
- e-checks, 27
- epidemic models, 68
- exponential quantile plot, 126
- exponentiality, 125
- File transfer, 92
- filesystems, 2
- FIS, 51, 69
- fixed information server, 51
- flow arrival count process, 142
- flow inter-arrival time, 142
- flow inter-arrivals, 142
- forwarding (FW), 51
- forwarding is enabled, 51
- Freenet, 8
- Gnutella, 8
- handoffs, 103
- hit ratios, 99
- hoarding, 4
- homeAP, 109
- hotspot, 123
- hybrid S-C, 51
- information dissemination, 173
- information locality, 102
- information providers, 11
- infostation, 46, 72
- infrastructure-wide modeling, 137
- inter-AP transition, 84
- inter-building transition, 84
- inter-building transitions, 109
- known-port limitation, 88
- Kolmogorov-Smirnov test, 124
- landmarks, 33
- link-level measurements, 180
- Location-aware services, 2
- location-based services, 1
- Location-sensing systems, 31
- Lognormal distribution, 132
- mesh networks, 5, 9, 78, 179
- micropayment, 27
- MIS, 51
- misclassified traffic, 88
- mobile client, 84
- mobile information access, 5
- mobile information server, 51
- mobile peer-to-peer, 10, 12
- mobile session, 84, 112
- mobile sessions, 111
- monitoring, 79, 88, 170
- monitoring tools, 79
- moving-average model, 155
- multi channel, 167
- multi-radio, 167
- multimedia traveling journal, 43
- Napster, 8
- network capacity, 167
- Network management, 92
- network-independent modeling, 129
- notesharing, 41
- ns-2, 53
- null hypothesis, 124
- on-line collaborations, 19
- one-state history, 118, 119
- P-P, 14, 50
- P-P variations, 51
- particle filter, 37
- passive querying, 22, 50
- peer-to-peer, 8, 9, 88, 92
- peer-to-peer caching, 96
- popular application, 9
- popular applications, 2, 92
- positioning systems, 31

- power law, 74
- power management, 169
- prefetching, 23
- quantile plot, 130
- queries, 22
- query, 51
- query interval, 53
- querying, 25
- Radar, 31
- random waypoint mobility, 52
- report, 22
- reusability, 137
- revisit, 115
- RF tags, 6
- roaming client, 84
- roaming session, 84
- Rosenstock's trapping model, 69
- S-C, 14, 50
- same-AP repeated requests, 96, 97
- same-building repeated requests, 96
- same-client repeated request, 96
- same-client repeated requests, 96
- sampling variability, 126
- scalability, 137
- scanning activity, 92
- security problems, 166
- self-similar traffic, 156
- sensors, 2
- session, 83, 111
- session arrivals, 142
- session duration, 111, 112
- sessions, 111, 129
- signature-based positioning, 31
- simple epidemic model, 68
- social network analysis, 19
- source-level modeling, 158
- spatial locality of information, 10, 94
- stationary session, 83, 112
- stationary sessions, 111
- stochastic order, 111, 112
- straight S-C, 51
- streaming, 2, 88
- streaming audio, 2
- subsequent request, 96
- synchronous mode, 14, 24, 50
- temporal locality, 96
- time-varying Poisson process, 122, 123
- token-based approaches, 27
- topology modeling, 164
- traffic load, 87
- traffic of APs, 84
- traffic share across applications, 92
- traffic variation, 140
- transient phenomena, 79
- transient sessions, 114
- trapping model, 69
- Ubisense, 31
- United Villages project, 10, 45
- uploaders, 87
- user cache, 96
- vehicle-based services, 2
- video, 2
- visit, 79, 83
- web, 9, 94
- web browsing, 92
- web requests, 96
- whiteboard, 41
- WiFi-enabled Kiosks, 10
- wireless infrastructure, 79
- wireless Internet, 11
- wireless measurements, 79
- Wireless PANs, 6
- Wireless WANs, 6
- wireless workload of APs, 101